

---

# **THE COMPARISON OF AC AND DC ALTERNATIVES FOR SUB-TRANSMISSION NETWORKS**

---

Frank Engelbrecht



Thesis presented in partial fulfilment of the requirements  
for the degree of Master of Science in Engineering at the  
University of Stellenbosch

Supervisor: Dr. H.J. Beukes

December 2001

## DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, unless otherwise stated, and has not previously, in its entirety or in part, been submitted at a university for a degree.

Frank Engelbrecht

8 October 2001



## SUMMARY

Recent advances in semiconductor technology extended the economic power range for DC transmission to just a few MW. Network planners need tools to compare AC and DC alternatives in order to find the best technical and economic solution for a specific network. TESAT, a software analysis tool, is developed to determine the optimum conductor and line technology for a network. Voltage regulation problems are identified and can be solved with network devices which have the potential to solve network problems more effectively and economically than ever before. PSAT, another software analysis tool developed in previous research, is used to model networks and support technologies. Hence, with the aid of TESAT and PSAT, line and support technologies are combined in an attempt to find the most effective solution in terms of cost and technical performance. This is demonstrated with the aid of a case study.

Furthermore, interfaces between PSAT and the real world are developed. This includes an extension to the input interface of PSAT that calculates the equivalent impedances of a transmission line automatically, as well as an interface to share data between ReticMaster and PSAT. A dispersed generation and support technology database is also developed as an extension to the output interface of PSAT.

## OPSOMMING

Onlangse vooruitgang in halfgeleiertegnologie het tot gevolg dat GS transmissie ekonomies is vir slegs 'n paar MW. Netwerkbepanners benodig gevolglik pakette om WS en GS alternatiewe te vergelyk vir 'n spesifieke netwerk. In hierdie tesis is 'n analitiese sagteware-pakket (TESAT) dus ontwikkel om die optimale geleier en lyntegnologie vir 'n netwerk te bepaal. Spanningsregulasie-probleme word geïdentifiseer en opgelos met netwerktoestelle wat die potensiaal het om netwerkprobleme meer doeltreffend en ekonomies as ooit tevore op te los. PSAT, 'n ander analitiese sagteware-pakket wat in vorige navorsing ontwikkel is, word dan ook gebruik om netwerke en steuningstegnologieë te modelleer. Dus word PSAT en TESAT gebruik om lyn- en steuningstegnologieë te kombineer. Die doel hiervan is om die mees doeltreffende oplossing in terme van kostes en tegniese werksverrigting te vind. Dit word met behulp van 'n gevallestudie gedemonstreer.

Verder word koppelvlakke tussen PSAT en die eksterne wêreld ontwikkel. Dit sluit in: (a) 'n uitbreiding van die intreekoppelvlak van PSAT wat die ekwivalente impedansie vir 'n transmissielyn outomaties bereken; (b) die koppelvlak om data te deel tussen PSAT en ReticMaster. 'n Verspreide generasie- en steuningstegnologie databasis is uiteindelik ook ontwikkel as 'n uitbreiding van die uittreekoppelvlak van PSAT.



TABLE OF CONTENTS

LIST OF FIGURES..... VI

LIST OF TABLES..... X

LIST OF ABBREVIATIONS..... XII

ACKNOWLEDGEMENTS .....XIV

CHAPTER 1 INTRODUCTION ..... 1

CHAPTER 2 METHOD OF ANALYSIS ..... 8

2.1 INTRODUCTION ..... 9

2.2 THE AC TRANSMISSION LINE MODEL ..... 10

2.2.1 Series impedance ..... 10

2.2.2 Shunt admittance ..... 21

2.2.3 AC transmission line models for short, medium and long lines ..... 25

2.2.4 Implementation ..... 30

2.3 THE DC TRANSMISSION LINE MODEL ..... 34

2.4 VOLTAGE REGULATION AND LINE LOSSES FOR AC TRANSMISSION LINES .. 36

2.4.1 Power flow in terms of ABCD constants ..... 36

2.4.2 Calculation of the sending-end voltage ..... 37

2.4.3 Calculation of the receiving-end voltage..... 41

2.4.4 Calculation of the sending-end power..... 42

2.5 VOLTAGE REGULATION AND LINE LOSSES FOR DC TRANSMISSION LINES .. 48

2.6 TRANSMISSION COSTS ..... 50

2.6.1 Review of the LCC ..... 50

2.6.2 Determination of the break-even distance..... 53

2.7 THE APPROPRIATE TECHNOLOGY INDEX (ATI) FOR AC AND DC LINE  
TECHNOLOGIES..... 55

2.7.1 ATI for AC line technologies ..... 55

2.7.2 ATI for DC line technologies..... 58

2.7.3 Implementation of the ATI..... 60

2.8 IMPLEMENTATION..... 63

CHAPTER 3 LOAD REACH AND TRANSFER CAPABILITIES OF AC  
AND DC LINE TECHNOLOGIES ..... 66

|  |  |            |
|--|--|------------|
| 3.1  | INTRODUCTION .....   | 67         |
| 3.2  | THERMAL LOAD REACH – A DETERMINISTIC AND PROBABILISTIC APPROACH<br>.....                     | 68         |
| 3.3  | THERMAL LOAD REACH OF AC AND DC LINE TECHNOLOGIES AT<br>DISTRIBUTION VOLTAGE LEVELS .....    | 72         |
| 3.4  | TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES AT<br>DISTRIBUTION VOLTAGE LEVELS ..... | 76         |
| <b>CHAPTER 4 THE KATIMA MULILO CASE STUDY .....</b>                    |  | <b>79</b>  |
| 4.1  | INTRODUCTION .....   | 80         |
| 4.2  | THE KATIMA MULILO CASE STUDY .....   | 81         |
| 4.2.1  | <i>AC line technology optimisation</i> .....   | 81         |
| 4.2.2  | <i>Obtaining network parameters</i> .....  | 88         |
| 4.2.3  | <i>Uncompensated network</i> .....   | 91         |
| 4.3  | SHUNT COMPENSATION .....   | 96         |
| 4.4  | SERIES COMPENSATION .....  | 102        |
| 4.5  | SERIES-SHUNT COMPENSATION.....   | 107        |
| 4.6  | IN-LINE COMPENSATION – THE DC ALTERNATIVE .....  | 110        |
| 4.6.1  | <i>DC line technology optimisation</i> .....   | 110        |
| 4.6.2  | <i>Simplified network</i> .....  | 115        |
| 4.7  | SUMMARY OF THE KATIMA MULILO CASE STUDY .....  | 120        |
| <b>CHAPTER 5 OTHER WORK AND FUTURE DEVELOPMENTS.....</b>               |  | <b>126</b> |
| 5.1  | PSAT – RETICMASTER INTERFACE .....   | 127        |
| 5.1.1  | <i>Implementation</i> .....  | 129        |
| 5.2  | SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE.....                                    | 132        |
| 5.3  | FUTURE DEVELOPMENTS .....  | 135        |
| <b>CHAPTER 6 CONCLUSION .....</b>                                      |  | <b>139</b> |
| <b>REFERENCES .....</b>  |  | <b>143</b> |
| <b>ADDENDUM A CONDUCTOR DATA AND CONVERTER STATION<br/>COSTS .....</b> |  | <b>148</b> |
| <b>ADDENDUM B PROGRAM LISTINGS .....</b>                               |  | <b>152</b> |



## LIST OF FIGURES

|   |    |
|---|----|
| Fig. 1.1: Different line and support technologies .....   | 5  |
| Fig. 2.1: Line section of length $\Delta x$ [B5] .....  | 10 |
| Fig. 2.2: Three-phase transmission line with earth replaced by earth return<br>conductors.....                    | 16 |
| Fig. 2.3: (a) Complete circuit and (b) simplified circuit representation of series-<br>phase impedances .....     | 18 |
| Fig. 2.4: Three-phase line with neutral conductors and with earth plane replaced<br>by image conductors .....     | 22 |
| Fig. 2.5: Representation of a two-port network [B5] .....   | 25 |
| Fig. 2.6: Short line [B5].....  | 26 |
| Fig. 2.7: Medium-length line - nominal $\pi$ circuit [B5].....  | 27 |
| Fig. 2.8: Overhead line section of length $\Delta x$ [B5].....  | 28 |
| Fig. 2.9: Overhead line equivalent $\pi$ circuit [B5] .....   | 30 |
| Fig. 2.10: Overhead line parameters interface.....  | 31 |
| Fig. 2.11: a) Add conductor and b) Modify conductor interface.....  | 32 |
| Fig. 2.12: a) Add tower and b) Modify tower interface .....   | 33 |
| Fig. 2.13: DC line section of length $\Delta x$ .....   | 34 |
| Fig. 2.14: DC transmission line model .....   | 35 |
| Fig. 2.15: Representation of a two-port network [B5] .....  | 36 |
| Fig. 2.16: DC transmission line model .....   | 48 |
| Fig. 2.17: A graphical representation of the relationship between $PV_{CIC}$ , $A_{MC}$ and<br>$CL(n)$ .....      | 51 |
| Fig. 2.18: KPIs for AC line technologies at various voltage levels: a) HV b) MV<br>and c) LV [C6].....            | 56 |
| Fig. 2.19: KPIs for DC line technologies at HV, MV and LV levels.....   | 59 |
| Fig. 2.20: Ranking for LCC evaluation .....   | 61 |
| Fig. 2.21: AC interface of TESAT .....  | 63 |
| Fig. 2.22: DC interface of TESAT .....  | 64 |
| Fig. 3.1: Deterministic thermal load reach of different Nampower conductors<br>from an AC and DC perspective..... | 73 |
| Fig. 3.2: Probabilistic thermal load reach of different Eskom conductors from an<br>AC and DC perspective .....   | 74 |



|  |    |
|--|----|
| Fig. 3.3: Probabilistic load reach of 33 kV three-phase and bipole Hare at different template temperatures .....   | 75 |
| Fig. 3.4: Line transfer capabilities for different Nampower AC and DC line technologies, distribution voltages and conductor sizes.....                            | 77 |
| Fig. 3.5: Line transfer capabilities for different Eskom AC and DC line technologies, distribution voltages and conductor sizes.....                               | 77 |
| Fig. 4.1: The Caprivi region .....   | 81 |
| Fig. 4.2: 132 kV Nampower line structures [C18] .....  | 82 |
| Fig. 4.3: Voltage regulation of the AC line technologies as a function of distance and load growth .....   | 83 |
| Fig. 4.4: Voltage regulation of the AC line technologies as a function of load growth.....   | 84 |
| Fig. 4.5: Voltage regulation of the AC line technologies as a function of position for an (a) initial load of 0.52 MVA and (b) a final load of 8.85 MVA .....      | 85 |
| Fig. 4.6: Contour plot for voltage regulation of the AC line technologies.....   | 85 |
| Fig. 4.7: Power losses of the AC line technologies as a function of load growth.....   | 86 |
| Fig. 4.8: LCC for the AC line technologies.....  | 87 |
| Fig. 4.9: ATI of the three-phase line technologies .....   | 88 |
| Fig. 4.10: Compensated source.....   | 89 |
| Fig. 4.11: Compensated source impedance and equivalent $\pi$ circuit model .....   | 90 |
| Fig. 4.12: Equivalent circuit diagram for the Katima Mulilo network.....   | 91 |
| Fig. 4.13: Voltage regulation as a function of network position and voltage angle for the uncompensated Katima Mulilo network .....                                | 92 |
| Fig. 4.14: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and an initial load of $5.21 \times 10^{-3}$ p.u. .... | 92 |
| Fig. 4.15: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and a final load of $88.54 \times 10^{-3}$ p.u. ....   | 93 |
| Fig. 4.16: LCC of the uncompensated Katima Mulilo network .....  | 94 |
| Fig. 4.17: Katima Mulilo network with a shunt compensator for light loads.....   | 97 |
| Fig. 4.18: Voltage plots at the PCC.....   | 97 |



Fig. 4.19: Active and reactive power dissipated across the shunt-compensated  
Katima Mulilo network ..... 98

Fig. 4.20: Power plots for the shunt compensator ..... 98

Fig. 4.21: Effect of purely reactive shunt compensation on the power losses of the  
uncompensated network ..... 99

Fig. 4.22: LCC of the uncompensated and shunt-compensated Katima Mulilo  
network (ALF = 0.4)..... 100

Fig. 4.23: LCC of the uncompensated and shunt-compensated Katima Mulilo  
network for an ALF of (a) 0.6 and (b) 0.2 ..... 101

Fig. 4.24: Katima Mulilo network with a series compensator..... 103

Fig. 4.25: Voltage magnitude and angle across the series-compensated Katima  
Mulilo network ..... 103

Fig. 4.26: Voltage plots at (a) PCCS and (b) PCCR..... 104

Fig. 4.27: (a) Voltage and (b) power plots for the series compensator ..... 104

Fig. 4.28: LCC of the uncompensated and series-compensated Katima Mulilo  
network ..... 106

Fig. 4.29: Katima Mulilo network with a series-shunt compensator..... 107

Fig. 4.30: Voltage plots at (a) PCCS and (b) PCCR..... 108

Fig. 4.31: True voltage injected by the series compensator ..... 108

Fig. 4.32: Power requirements of the (a) series and (b) shunt devices..... 109

Fig. 4.33: 66 kV Nampower tower (Kamerad) [C18] ..... 111

Fig. 4.34: Voltage regulation of the DC line technologies as a function of load  
growth..... 112

Fig. 4.35: Power losses of the DC line technologies as a function of load growth... 113

Fig. 4.36: LCC for the DC line technologies..... 114

Fig. 4.37: ATI of the bipolar line technologies ..... 115

Fig. 4.38: Katima Mulilo network with an in-line compensator ..... 115

Fig. 4.39: Voltage plots at (a) PCCS and (b) PCCR for an initial load of  
 $5.21 \times 10^{-3}$  p.u. .... 116

Fig. 4.40: Power requirements of the device at (a) PCCS and (b) PCCR for an  
initial load of  $5.21 \times 10^{-3}$  p.u. .... 117

Fig. 4.41: Voltage plots at (a) PCCS and (b) PCCR for a final load of  
 $88.54 \times 10^{-3}$  p.u. .... 117

Fig. 4.42: Power requirements of the device at (a) PCCS and (b) PCCR for a final  
load of  $88.54 \times 10^{-3}$  p.u..... 118

Fig. 4.43: Break-even distances for (a) final load conditions and (b) initial load  
conditions..... 119

Fig. 4.44:  $PV_{LCC}$  for the different network solutions ..... 124

Fig. 5.1: Modelling of a reticulation network..... 128

Fig. 5.2: *User Report Page* tab in ReticMaster ..... 130

Fig. 5.3: The user report in ReticMaster..... 130

Fig. 5.4: Interface to extract network parameters from ReticMaster..... 131

Fig. 5.5: Device library window ..... 132

Fig. 5.6: a) Add device and b) modify device interfaces..... 134

Fig. 5.7: Voltage regulation of 76.21 kV SWER lines as a function of load growth 138

Fig. 6.1: Inputs and outputs of the TESAT engine ..... 140



## LIST OF TABLES

|  |     |
|--|-----|
| Table 2-1: Temperature coefficient of resistance at 20°C [C4].....   | 12  |
| Table 2-2: Skin effect table [B6] .....  | 13  |
| Table 2-3: Resistivity of different soils [B10].....   | 17  |
| Table 2-4: Summary of Overhead Line Parameters [B5].....   | 30  |
| Table 3-1: Conductor ampacity table using deterministic loading techniques<br>[C12].....   | 69  |
| Table 3-2: Conductor ampacity table using probabilistic loading techniques<br>[C13], [C12].....                                      | 70  |
| Table 4-1: 132 kV Nampower line technology data.....   | 82  |
| Table 4-2: Summary of the voltage regulation performance.....  | 84  |
| Table 4-3: Summary of power losses .....   | 86  |
| Table 4-4: Summary of LCC and cost of losses .....   | 87  |
| Table 4-5: ATI summary of the three-phase line technologies .....  | 88  |
| Table 4-6: Source impedances.....  | 89  |
| Table 4-7: Additional capital costs incurred for the uncompensated Katima<br>Mulilo network .....                                    | 94  |
| Table 4-8: Summary of the LCC of the uncompensated Katima Mulilo network .....   | 94  |
| Table 4-9: Summary of total power losses for uncompensated and<br>shunt-compensated Katima Mulilo network.....                       | 100 |
| Table 4-10: Summary of the LCC of the uncompensated and shunt-compensated<br>Katima Mulilo network for different values of ALF ..... | 101 |
| Table 4-11: Additional capital costs incurred for the series-compensated Katima<br>Mulilo network .....                              | 105 |
| Table 4-12: Summary of the LCC of the uncompensated and series-compensated<br>Katima Mulilo network .....                            | 106 |
| Table 4-13: ±50 kV bipolar DC line data .....  | 111 |
| Table 4-14: Summary of voltage regulation performance.....   | 112 |
| Table 4-15: Summary of power losses .....  | 113 |
| Table 4-16: Summary of LCC and cost of losses .....  | 114 |
| Table 4-17: ATI summary of the bipolar line technologies .....   | 115 |
| Table 4-18: Summary of the line technology performance in the Katima Mulilo<br>network .....   | 120 |

Table 4-19: Summary of the voltage regulator performance in the Katima Mulilo network ..... 122

Table 4-20: Summary of the economic analysis for the Katima Mulilo case study.. 124

Table 5-1: Summary of the voltage regulation performance..... 138

Table A-1: Conductor Data [C4] ..... 149

Table A-2: Estimated converter station costs ..... 150

## LIST OF ABBREVIATIONS

|       |  |
|-------|--|
| AAAC  | All-Aluminium-Alloy Conductor              |
| AAC   | All-Aluminium Conductor                    |
| AACSR | Aluminium-Alloy Conductor Steel-Reinforced |
| ABC   | Aerial Bundled Conductor                   |
| AC    | Alternating Current                        |
| ACAR  | Aluminium Conductor Alloy-Reinforced       |
| ACSR  | Aluminium Conductor Steel-Reinforced       |
| ALF   | Annual Load Factor                         |
| ATI   | Appropriate Technology Index               |
| CIC   | Capital Investment Cost                    |
| CL    | Cost of Losses                             |
| DC    | Direct Current                             |
| DVR   | Dynamic Voltage Restorer                   |
| ELF   | Energy Loss Factor                         |
| EVR   | Electronic Voltage Regulator               |
| FACTS | Flexible AC Transmission Systems           |
| GC    | Generation Cost                            |
| GMD   | Geometric Mean Distance                    |
| GMR   | Geometric Mean Radius                      |
| GUI   | Graphical User Interface                   |
| HV    | High-Voltage                               |
| IGBT  | Insulated Gate Bipolar Transistor          |
| KPI   | Key Performance Indicator                  |
| LCC   | Life-Cycle Cost                            |
| LV    | Low-Voltage                                |
| MV    | Medium-Voltage                             |
| OLP   | Overhead Line Parameters                   |
| PCC   | Point of Common Coupling                   |
| PCCL  | Point of Common Coupling at Load           |
| PCCR  | Point of Common Coupling at Receiving end  |
| PCCS  | Point of Common Coupling at Sending end    |
| PSAT  | Power System Analysis Tool                 |



|         |   |
|---------|---|
| PVM     | Present Value of Money                          |
| SIL     | Surge Impedance Load                            |
| SSSC    | Static Synchronous Series Compensator           |
| STATCOM | Static Synchronous Compensator                  |
| SVC     | Static Var Compensator                          |
| SWER    | Single Wire Earth Return                        |
| TCSC    | Thyristor Controlled Series Capacitor           |
| TESAT   | Technical and Economical Software Analysis Tool |
| UPFC    | Unified Power Flow Controller                   |
| VSC     | Voltage Source Converter                        |

## **ACKNOWLEDGEMENTS**

I would like to convey my sincere gratitude to the following people and institutions that made this thesis possible.

Firstly, to Dr. Johan Beukes for his guidance and invaluable support.

To Nampower for their financial support and to allow me the opportunity for further studies.

To the members of the power electronics group who made it enjoyable and rewarding to be part of the group.

And of course to all my friends and family, especially my parents, for their patience, support and understanding.

Finally I would like to thank the Almighty and Heavenly Father for spiritual guidance and support.

# CHAPTER 1 INTRODUCTION

A power transmission system serves a dual purpose, i.e. system interconnection and bulk energy transfer. Total power generation capacity and fuel cost can be minimised using a transmission system to pool power plants and load centres. Transmission interconnections enable one to take advantage of diversity of loads, availability of sources and fuel price in order to supply electricity to loads at minimum cost with a required reliability [B1].

There are various alternatives as far as bulk energy transfer is concerned, not all of them involving electric power transmission and an economic assessment is essential in each case. Whenever the transmission distance is sufficiently large, and restricting the choice to the electrical alternatives, the case of high voltage direct current (HVDC) transmission is well established in spite of the relatively high cost of the dual conversion required [B2]. The use of the Insulated Gate Bipolar Transistor (IGBT) in DC transmission, however, has extended the economic power range for DC transmission to just a few MW.

In less obvious decisions, such as schemes to increase the power transmission capability inside of a system or to transmit a small amount of power over a large distance, the economic comparison between alternating current (AC) and DC must include the cost of lines, terminals, any special apparatus needed for voltage support, short circuit limitation, etc. The energy losses and the plant needed to supply it must also be capitalised.

To achieve a meaningful and generally applicable comparison between AC and DC in marginal cases is indeed a very difficult task. Among the factors responsible for the complexity of a generalised theory are [B2]:

- The wide range of practical situations involving different conditions among countries, e.g. the costs of overhead lines vary from country to country by a factor as high as 2.5, while the cost of the converter terminals varies very little;
- The lack of technical comparability, given the rather different degrees of AC and AC-DC power systems;



- The need to consider the long-term effects on overall system design and cost when choosing among alternative plans for system development;
- The rapid development made in the technology of both AC and DC transmission. In this respect transmission line costs have experienced a large increase in recent years. Since the line cost is relatively lower in the case of DC transmission, this effect has affected the AC alternative. However, the appearance of Flexible AC Transmission System (FACTS) devices is now having a similar beneficial effect on AC transmission costs.

Network planners need tools to compare AC and DC alternatives in order to find the best technical and economic solution when planning new networks or upgrading existing networks. An analysis tool would provide planners with the ability to compare AC and DC transmission line technologies in a manner known to power system engineers, i.e. with loadflow analysis and financial comparisons between different line options. The two most important characteristics of this tool must be:

- Assistance in the understanding of AC and DC as a transmission technology and the problems associated with these technologies;
- Provision of an analysis tool for first assessment of the voltage regulation, line losses and life-cycle costs (LCC) of different conductors from an AC and DC perspective in order to select the optimum conductor for the task at hand.

Such an analysis tool was developed under the name TESAT (Technical and Economical Software Analysis Tool). TESAT allows network planners to evaluate different AC and DC line options on a per project basis. The optimum conductor for a particular AC or DC line technology is identified, as well as any voltage regulation problems that will require some form of line compensation. Apart from being an analysis tool, TESAT can also be used as a training tool to educate network planners in the use of AC and DC as a transmission technology.

Network technologies present an alternative in terms of network devices, which can be used to solve network problems. Recent advances in semiconductor technology pushed a large number of network devices onto the market. These devices have the potential to solve network problems, including voltage regulation, more effectively



and economically than ever before [A1]. All network support devices can be classified into one of four categories according to their configuration, i.e. shunt, series, series-shunt and in-line devices. DC transmission is an in-line technology that connects two separate AC networks by a DC link. DC is, therefore, not a grid network in the way that an AC system is, nor is it expected to be. The role of DC transmission, for economic reasons, is to interconnect AC systems where a reliable AC interconnection would be too expensive [B3].

When constructing new networks, utility planners are frequently reluctant to integrate network support technologies with the planning process, mainly due to a lack of adequate planning tools. To address these needs a software analysis tool was developed to model networks and support technologies. The aim of PSAT (Power System Analysis Tool) is to assist in the understanding of network problems, network technologies and their interaction and to provide an analysis tool for first assessment in the network problem-solving process [A2].

When planning a new network or upgrading an existing network, the aim of TESAT is to assist in the selection of:

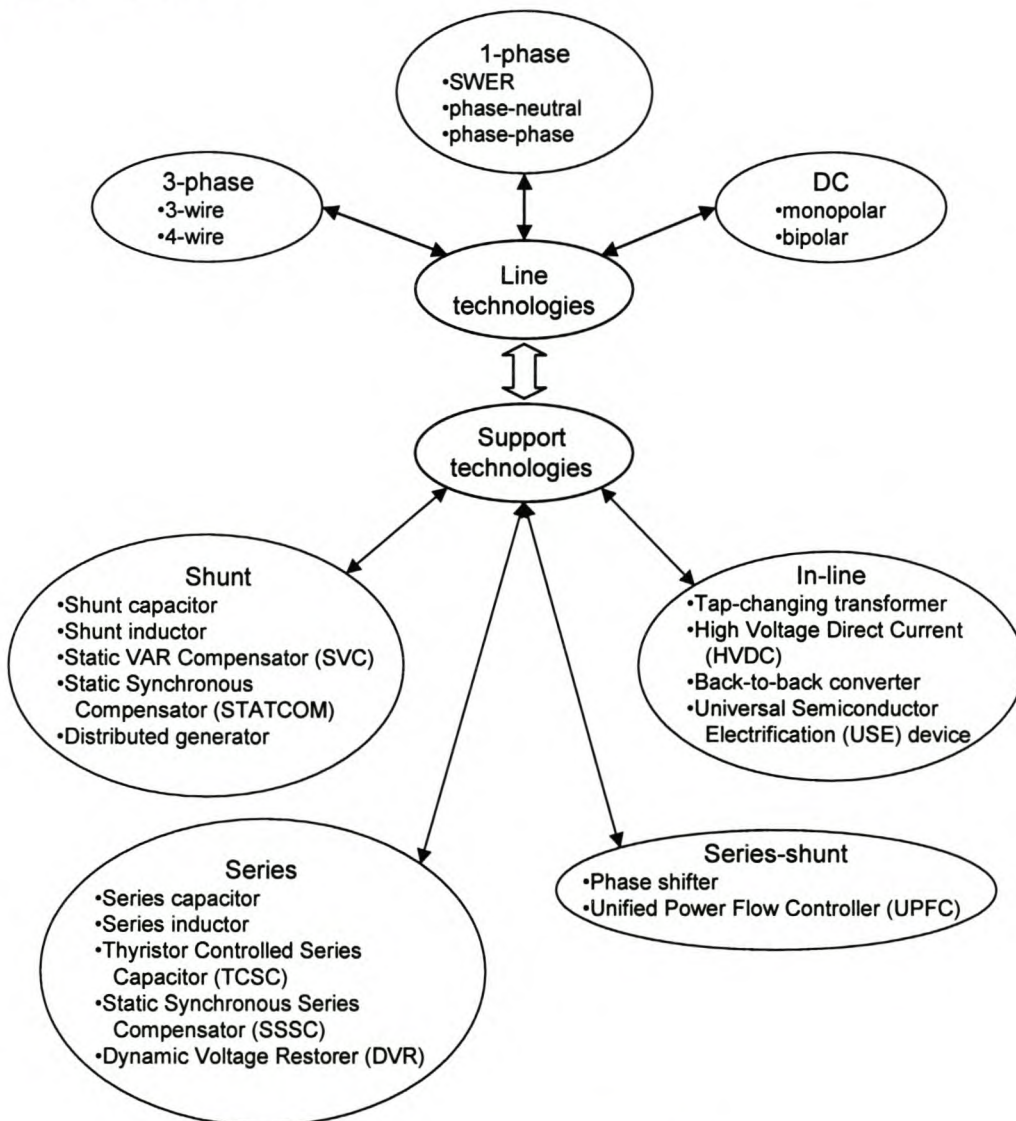
1. The line technology for the network;
2. The optimum conductor for the line technology;
3. Identification of any voltage regulation problems.

After determining the three aspects above with the aid of TESAT, PSAT can subsequently be used to analyse the complete network. Voltage regulation problems in the new networks can be identified and solved in the design phase, resulting in the construction of an already compensated network. This can result in extensive cost benefits as post-construction alternations to the network are eliminated. The economic impact of the various solutions to a network problem can be evaluated in TESAT with various conductors for a particular line technology, in order to select the most effective solution, both in technical and financial terms.

It is important to note the difference between TESAT, PSAT and other simulation packages. When designing new networks TESAT can be used to determine the optimum conductor for a particular line technology and identify any voltage

regulation problems. PSAT can then be used to analyse the complete network, evaluating all possible solutions and identifying one or two optimum solutions, which can quickly be evaluated in TESAT to judge the cost implications of the various solutions to the network problem. These optimum solutions can then be studied in detail in existing packages such as PSS/U and ReticMaster.

Presently only three-phase and bipolar DC lines are considered in TESAT. Future developments in TESAT will include alternative line technologies such as SWER (single wire earth return), single-phase and phase-phase networks, as well as monopolar DC lines.



**Fig. 1.1: Different line and support technologies**

In today's economic environment it is of the utmost importance that optimum use is made of capital and resources when planning new networks or upgrading existing



networks. The aim of this thesis is to show how the network planner can integrate line and support technologies in the planning process to determine the most effective solution in terms of cost and technical performance. This will allow the planner to select a conductor and line technology that will meet the power requirements of the network. Network support technologies can then subsequently be used to solve any voltage regulation problems in an attempt to stretch networks up to thermal capacity. The different line and support technologies are shown in Fig. 1.1. Secondly, the aim of this thesis is to document the development and practical application of the three-phase AC and bipolar DC engines of TESAT.

In the development of TESAT, the following contributions were made:

1. The process of determining the optimum conductor for different AC and DC transmission technologies was documented extensively on a high level.
2. Equations were documented to calculate the voltage regulation, line losses and transmission costs of AC transmission lines. Similar equations were developed for the voltage regulation, line losses and transmission costs of DC transmission lines.
3. A technology index was developed to compare various DC line technologies on a per project basis. The technology index for AC line technologies with and without network support devices were also documented.
4. Three-dimensional graphs were created to portray the voltage regulation, line losses, transmission costs and technology indexes of different AC and DC line technologies for all values of line position and point load.

Other contributions of this thesis include:

1. Comparisons between the load reaches and transfer capabilities of AC and DC lines at distribution voltage levels.
2. The development of interfaces between PSAT and the real world. This included the development of the line parameters interface, the ReticMaster interface and the support technology and dispersed generation database.

Chapter 2 discusses the process of determining the optimum conductor for AC and DC line technologies on a high level. The AC and DC transmission line models are discussed extensively and a software tool to calculate the circuit parameters for the



AC model is presented. Special attention is devoted to the voltage regulation, power losses and costs of AC and DC line technologies. An index is also developed to compare objectively various line technologies on a per project basis. Chapter 2 is concluded with a discussion of the TESAT interface, its features and its development.

Chapter 3 focuses on AC and DC line technologies at distribution voltage levels. The chapter starts with the deterministic and probabilistic determination of conductor ampacity ratings for both AC and DC transmission. The thermal load reach capabilities of three-phase and bipolar line technologies are calculated next at different voltage regulation margins for distribution networks. The chapter is concluded with the calculation of the transfer capabilities of these line technologies at different voltage regulation margins.

The Katima Mulilo case study is presented in Chapter 4. The study involves the construction of a new line to transmit a small amount of power over a large distance. Firstly, TESAT is used to determine the optimum conductor for a three-phase AC line. Voltage regulation of the optimised line with the transformer and load impedances is then subsequently analysed in PSAT with shunt, series and series-shunt compensators. DC transmission is implemented in PSAT as an in-line technology where TESAT determines the optimum conductor to be used for the DC link. Critical evaluation of the voltage regulation, power losses and transmission costs graphs also demonstrates the tutoring ability of TESAT. The chapter is concluded with a discussion of the most effective solution in terms of cost and technical performance.

Chapter 5 discusses the development of a PSAT interface to extract the electrical characteristics of a reticulation network from ReticMaster. The development of a support technology and dispersed generation database is also presented. Future developments for the overhead line parameters interface of PSAT and TESAT is proposed, as well as possibilities for future studies.

The thesis is concluded in Chapter 6 with a summary of the theoretical and practical results obtained with PSAT and TESAT.

## **CHAPTER 2 METHOD OF ANALYSIS**

### **2.1 INTRODUCTION**

### **2.2 THE AC TRANSMISSION LINE MODEL**

#### **2.2.1 Series impedance**

#### **2.2.2 Shunt admittance**

#### **2.2.3 AC transmission line models for short, medium and long lines**

#### **2.2.4 Implementation**

### **2.3 THE DC TRANSMISSION LINE MODEL**

### **2.4 VOLTAGE REGULATION AND LINE LOSSES FOR AC TRANSMISSION LINES**

#### **2.4.1 Power flow in terms of ABCD constants**

#### **2.4.2 Calculation of the sending-end voltage**

#### **2.4.3 Calculation of the receiving-end voltage**

#### **2.4.4 Calculation of the sending-end power**

### **2.5 VOLTAGE REGULATION AND LINE LOSSES FOR DC TRANSMISSION LINES**

### **2.6 TRANSMISSION COSTS**

#### **2.6.1 Review of the LCC**

#### **2.6.2 Determination of the break-even distance**

### **2.7 THE APPROPRIATE TECHNOLOGY INDEX (ATI) FOR AC AND DC LINE TECHNOLOGIES**

#### **2.7.1 ATI for AC line technologies**

#### **2.7.2 ATI for DC line technologies**

#### **2.7.3 Implementation of the ATI**

### **2.8 IMPLEMENTATION**



## 2.1 INTRODUCTION

This chapter discusses the development of methods for technical and economic comparisons between AC and DC transmission. The AC transmission line model is discussed first, after which section 2.3 deals with the DC transmission line model. In sections 2.4 and 2.5 the voltage regulation and line losses for AC and DC lines are discussed, followed by a discussion on transmission costs in section 2.6. The ATI models for AC and DC line technologies are presented in section 2.7. A discussion of TESAT concludes the chapter.

Transmission lines transmit electrical power from a sending end to a receiving end without supplying any consumers. In contrast, a distribution line supplies consumers directly at short intervals along the line. In practice the distinction between transmission and distribution is not so clear-cut as several large consumers are now being supplied at 132 kV and above [B4]. For the purposes of this discussion transmission voltages are assumed to be above 132 kV, sub-transmission voltages between 132 kV and 33 kV and distribution voltages 33 kV and less.

Low-voltage (LV) distribution networks operate up to and including voltages of a 1 000 V r.m.s., while medium-voltage (MV) distribution networks operate at voltages exceeding 1 000 V r.m.s. up to and including 33 kV. High-voltage (HV) networks operate at voltages above 33 kV.

## 2.2 THE AC TRANSMISSION LINE MODEL

Fig. 2.1 represents a line section of length  $\Delta x$ . Conductor spacing, types, and sizes determine the series impedance and shunt admittance. Series impedance affects line-voltage drops,  $I^2R$  losses and stability limits. Shunt admittance, primarily capacitive, affects line-charging currents, which inject reactive power into the power system [B4].

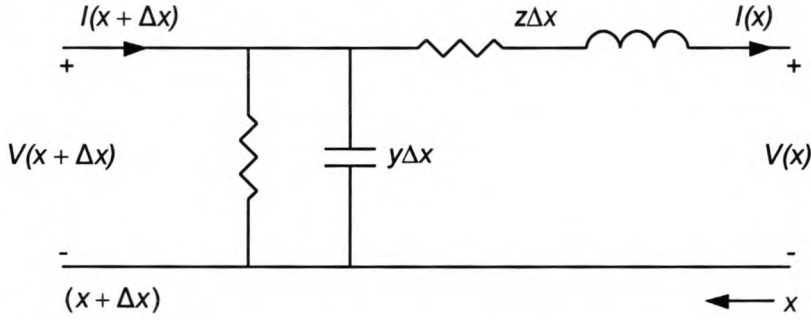


Fig. 2.1: Line section of length  $\Delta x$  [B5]

The matrices to calculate the series impedance and shunt admittance for untransposed as well as transposed three-phase lines with neutral conductors are developed in sections 2.2.1 and 2.2.2, followed by a discussion on the transmission line models for short, medium and long lines in section 2.2.3. Section 2.2 is concluded with a discussion on the line parameter interface of PSAT.

### 2.2.1 Series impedance

#### Series resistance

The content of this section is based on work done by Smit [C3].

The total AC resistance ( $R_{ac}$ ) of a conductor can be calculated as [B5]:

$$R_{ac} = \frac{P_{loss}}{|I|^2} \quad [\Omega] \quad (2-1)$$

where  $P_{loss}$  is the conductor real power loss and  $I$  is the rms conductor current. However, many times  $P_{loss}$  and  $I$  are unknown and must be calculated. The manufacturers of overhead conductors supply only the DC resistance per kilometre ( $r_{dc}$ ) at 20°C, which includes the spiralling effect for stranded conductors. The AC



resistance per kilometre ( $r_{ac}$ ) can be calculated in one of two ways and will be discussed next.

### ***Method 1: Various effects***

$r_{ac}$  can be calculated as follows:

- a) Obtain  $r_{dc}$  at 20°C from the data sheets for overhead aluminium conductors.
- b)  $r_{dc}$  is then corrected to the required operating temperature, e.g. at 40°C.
- c) The new  $r_{dc}$  is corrected for the skin effect.
- d) The resistance of aluminum conductor, steel-reinforced (ACSR) conductors should be further corrected for hysteresis and eddy current losses.
- e)  $r_{ac}$  is also dependent on the proximity effect of conductors.

### ***Method 2: Empirical formula***

It is difficult to calculate all the effects (c) – (e) in method 1. An alternative method can be used:

- f) Empirical formulae to calculate  $r_{ac}$ .

Steps (a) – (f) will now be discussed in detail in the following sections. All-aluminium-alloy conductor (AAAC) and all-aluminium conductor (AAC) resistances should be calculated according to Method 1, with only the skin effect taken into account. ACSR conductor resistances should be calculated according to Method 2, with hysteresis and eddy current losses taken into account.

- a)  $r_{dc}$  at 20°C can be obtained from the Aberdare tables [C4].
- b) The variation of temperature is considered as linear over the operating range, although it is not linear over a wide range of temperatures.  $r_{ac}$  can only be calculated if  $r_{dc}$  is known at the operating temperature. The following method is used:

$$R_2 = R_1 + \alpha R_1 (T_{av} - 20) \quad (2-2)$$

where  $R_1$  = DC resistance at temperature  $T_1$  [°C]

$R_2$  = DC resistance at temperature  $T_2$  [°C]



$$\begin{aligned}\alpha &= \text{temperature coefficient of resistance at } 20^{\circ}\text{C} \\ T_{av} &= \text{average temperature } [^{\circ}\text{C}]\end{aligned}$$

The manufacturers of overhead conductors supply the  $\alpha$  values for different conductors, given in Table 2-1 [C4]. There is a difference between the core and surface temperatures of a conductor, but this is only between  $0.2^{\circ}\text{C}$  and  $\pm 4^{\circ}\text{C}$ . The average temperature,  $T_{av}$ , is therefore assumed the same as the surface temperature.

**Table 2-1: Temperature coefficient of resistance at  $20^{\circ}\text{C}$  [C4]**

| Conductor type       | $\alpha$ |
|----------------------|----------|
| Copper               | 0.00393  |
| Cadmium copper       | 0.00300  |
| Hard-drawn aluminium | 0.00403  |
| Aluminium alloy      | 0.00360  |

- c) The resistance of non-magnetic conductors varies with frequency. This is caused by current flowing nearer to the outer surface of the conductor as a result of non-uniform flux distribution in the conductor. This phenomenon is commonly known as the skin effect and must be taken into account.  $r_{ac}$  at the desired operating temperature is calculated to include the skin effect as follows:

$$r_{ac} = Kr_{dc} \quad [\Omega/\text{km}] \quad (2-3)$$

$$\begin{aligned}\text{where } r_{ac} &= \text{AC resistance at the desired frequency} \\ r_{dc} &= \text{DC resistance at the desired temperature} \\ K &= \text{value in Table 2-2}\end{aligned}$$

In Table 2-2,  $K$  is a function of  $X$ , where  $K$  is interpolated for a specific  $X$  value:

$$X = 0.050138 \sqrt{\frac{\mu f}{r_{dc}}} \quad (2-4)$$

$$\begin{aligned}\text{where } f &= \text{frequency [Hz]} \\ \mu &= \text{relative permeability (1 for non-magnetic materials)}\end{aligned}$$

**Table 2-2: Skin effect table [B6]**

| $X$ | $K$     | $X$ | $K$     | $X$ | $K$     | $X$ | $K$     |
|-----|---------|-----|---------|-----|---------|-----|---------|
| 0.0 | 1.00000 | 1.0 | 1.00519 | 2.0 | 1.07816 | 3.0 | 1.31809 |
| 0.1 | 1.00000 | 1.1 | 1.00758 | 2.1 | 1.09375 | 3.1 | 1.35102 |
| 0.2 | 1.00001 | 1.2 | 1.01071 | 2.2 | 1.11126 | 3.2 | 1.38504 |
| 0.3 | 1.00004 | 1.3 | 1.01470 | 2.3 | 1.13069 | 3.3 | 1.41999 |
| 0.4 | 1.00013 | 1.4 | 1.01969 | 2.4 | 1.15207 | 3.4 | 1.45570 |
| 0.5 | 1.00032 | 1.5 | 1.02582 | 2.5 | 1.17538 | 3.5 | 1.49202 |
| 0.6 | 1.00067 | 1.6 | 1.03323 | 2.6 | 1.20056 | 3.6 | 1.52879 |
| 0.7 | 1.00124 | 1.7 | 1.04205 | 2.7 | 1.22753 | 3.7 | 1.56587 |
| 0.8 | 1.00212 | 1.8 | 1.05240 | 2.8 | 1.25620 | 3.8 | 1.60314 |
| 0.9 | 1.00340 | 1.9 | 1.06440 | 2.9 | 1.28644 | 3.9 | 1.64051 |

- d) The resistance of ACSR conductors is affected by the hysteresis in the steel core, where the number of layers plays a role. The current spirals around the core, following the strands. The magnetising effect is opposite for the layers in a two-layer conductor and results in a small increase of the effective resistance. The biggest increase is for a single layer, up to 15%. The resistance for a three-layer conductor increases by 5% to 10% at the rated full load of the conductor.

Eddy currents are produced due to cyclic magnetisation and cause heating. The heating effect is usually negligible for non-ferrous conductors. The total heat gain of the conductor consists of an eddy current and a hysteresis component, where the hysteresis component is about 2 – 5 times the eddy current component. These values are difficult to determine and the empirical formula of (f) offers an easier solution.

- e) The proximity effect is negligible for:

$$\frac{s}{d} > 5 \quad (2-5)$$

where  $s$  is the spacing between the conductors and  $d$  the overall diameter of the conductor [A3]. For a ratio  $s/d = 3$ , the proximity effect causes an increase in resistance of less than 1% [C5]. Also, the spacing between phases is usually more



than 5 times the conductor diameter, which means that the proximity effect may be ignored.

- f) An empirical formula is presented as an alternative method to calculate the skin, hysteresis and proximity effects [A4]. The theory works on the basis that the power input must be the same for both AC and DC circuits for the same average temperature of the conductor:

$$\begin{aligned} I_{ac}^2 r_{ac} &= I_{dc}^2 r_{dc} \\ \Rightarrow r_{ac} &= \left( \frac{I_{dc}}{I_{ac}} \right)^2 r_{dc} \end{aligned} \quad (2-6)$$

The DC current that will result in a certain temperature being reached is calculated and the empirical formulae are then used to convert the DC to the AC current:

- i) For ACSR conductors with three layers of aluminium strands:

$$I_{ac} = I_{dc} / \sqrt{(1.0123 + 2.319 \cdot 10^{-5} I_{dc})} \quad (2-7)$$

- ii) For ACSR conductors with single- and double-layer aluminium strands and a nominal aluminium area of  $\geq 175 \text{ mm}^2$ :

$$I_{ac} = I_{dc} / \sqrt{(1.0045 + 0.09 \cdot 10^{-6} I_{dc})} \quad (2-8)$$

- iii) For other ACSR conductors with single- and double-layer aluminium strands and a nominal aluminium area of  $< 175 \text{ mm}^2$ :

$$I_k = \frac{I_{dc}}{a} \quad (2-9)$$

where  $a$  is the nominal aluminium area and  $I_k$  the DC current per  $\text{mm}^2$ .

If  $I_k \leq 0.742$  then

$$I_{ac} = I_{dc} \quad (2-10)$$

If  $0.742 < I_k \leq 2.486$  then

$$I_{ac} = I_{dc} / \sqrt{\left(1 + 0.5142 - 2.678I_k + 5.772I_k^2 - 6.69I_k^3 + 4.53I_k^4 - 1.7946I_k^5 + 0.3862I_k^6 - 0.03488I_k^7\right)} \quad (2-11)$$

If  $2.486 < I_k \leq 3.908$  then

$$I_{ac} = I_{dc} / \sqrt{\left(1 + 0.0596 - 0.4404I_k + 0.4974I_k^2 - 0.2328I_k^3 + 0.0595I_k^4 - 0.0083I_k^5 + 0.0005I_k^6\right)} \quad (2-12)$$

If  $I_k > 3.908$  then

$$I_{ac} = \frac{I_{dc}}{\sqrt{1.1}} \quad (2-13)$$

For bundled conductors the effective resistance per unit length of the bundled conductor is equal to  $r_{ac}/n$ , where  $n$  is the amount of conductors per bundle ( $n = 2, 3$ , or  $4$ ).

Line resistance is dependent on the chosen operating temperature. The system planner uses a single value of line resistance in all power network simulations. For clearance purposes, most of the lines are designed for an operating temperature of only  $50^\circ\text{C}$ . Probability studies will show that the conductor is operated at temperatures below  $50^\circ\text{C}$  for most of the time. Line resistances should, therefore, be calculated at an operating temperature  $\leq 50^\circ\text{C}$ . It is recommended that all resistances be calculated at the temperature of  $40^\circ\text{C}$ .

### Series inductance

The line inductance depends on the partial flux linkages within the conductor cross-section and external flux linkages [B7]. For a completely transposed three-phase, three-wire line with equal phase spacing, the inductance per phase is:

$$L_1 = L_2 = 2 \times 10^{-4} \ln \frac{D_{eq}}{D_s} \quad [\text{H/km}] \quad (2-14)$$

where

$$GMD = D_{eq} = \sqrt[3]{d_{12}d_{23}d_{31}} \quad (2-15)$$

is the geometric mean distance (*GMD*) between the conductors of phases *a*, *b*, and *c*. In the above equation  $D_S$  is the geometric mean radius (*GMR*) of the phase conductors.

To calculate  $L_1 = L_2$  for bundled conductors with bundle spacing *d*, an equivalent conductor with  $GMR = D_{SL}$  replaces the bundle.  $D_{SL}$  replaces  $D_S$  in equation (2-14) and is calculated as follows:

Two-conductor bundle (twin):

$$D_{SL} = \sqrt{D_S d} \quad (2-16)$$

Three-conductor bundle (triple):

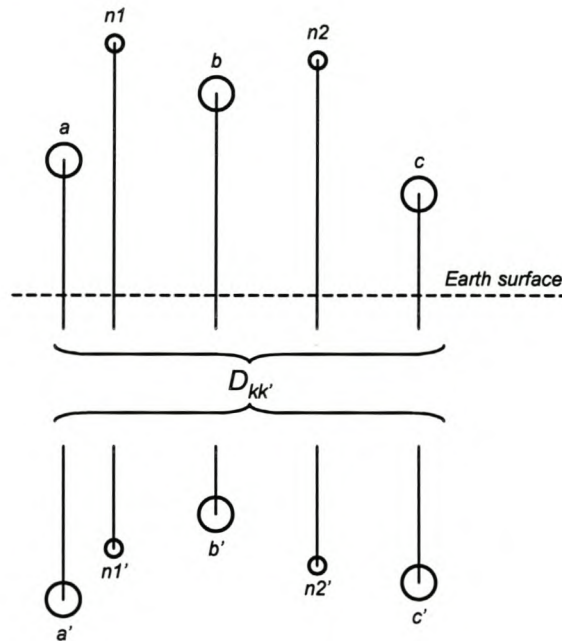
$$D_{SL} = \sqrt[3]{D_S d^2} \quad (2-17)$$

Four-conductor bundle (quad):

$$D_{SL} = 1.091 \sqrt[4]{D_S d^3} \quad (2-18)$$

## Series impedance

In this section equations are developed for calculation of the series impedance of the three-phase line shown in Fig. 2.2.



**Fig. 2.2: Three-phase transmission line with earth replaced by earth return conductors**

The line has three phase conductors *a*, *b*, and *c*, where bundled conductors, if any, have already been replaced by the equivalent conductors, described in equations



(2-16) - (2-18). The line also has two neutral conductors. All neutral conductors are connected in parallel and are grounded to the earth at regular intervals along the line. Any isolated neutral conductors that carry no current are omitted. The phase conductors are isolated from each other and from earth [B5].

If the phase currents are not balanced, there may be a return current in the grounded neutral wires and in the earth. The earth return current will spread out under the line, seeking the lowest impedance return path. A classic paper by Carson [A5], later modified by others [B8], [B9], shows that the earth can be replaced by a set of equivalent “earth return” conductors located directly under the overhead conductors, as shown in Fig. 2.2. Each earth return conductor carries the negative of its conductor current, has a *GMR* denoted  $D_{k'k'}$ , distance  $D_{kk'}$  from its overhead conductor, and resistance  $r_{k'}$  given by:

$$D_{k'k'} = D_{kk} \quad [\text{m}] \quad (2-19)$$

$$D_{kk'} = 658.5 \sqrt{\frac{\rho}{f}} \quad [\text{m}] \quad (2-20)$$

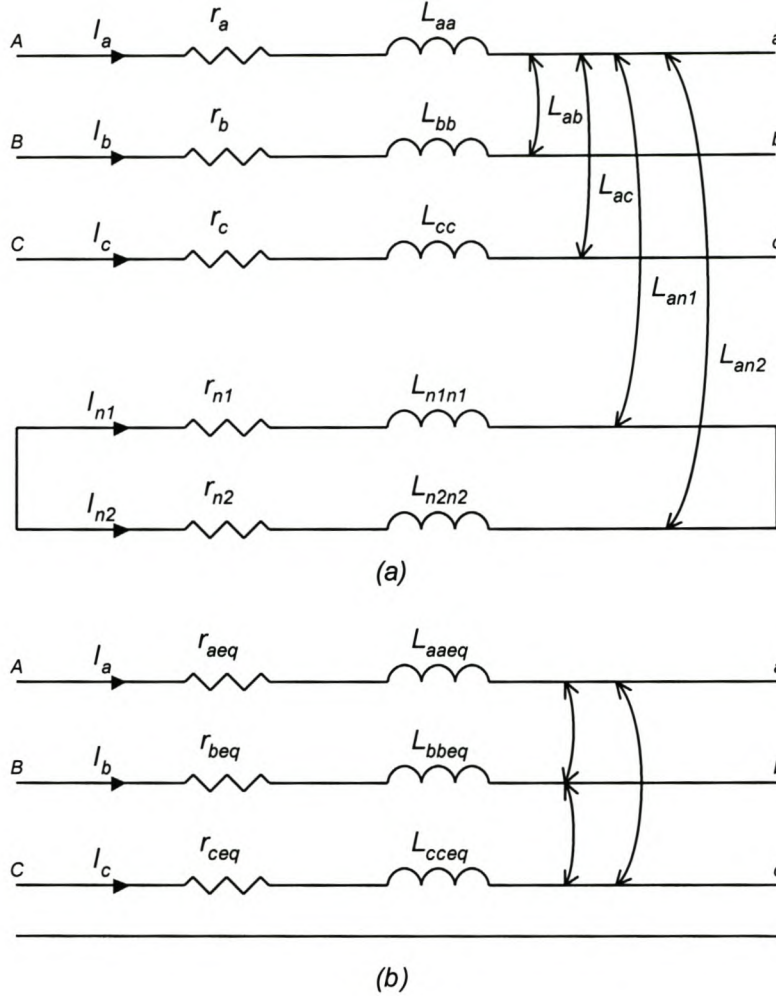
$$r_{k'} = 9.869 \times 10^{-4} \times f \quad [\Omega/\text{km}] \quad (2-21)$$

where  $\rho$  is the earth resistivity and  $f$  the frequency. The value of  $\rho$  for various earth types is given in Table 2-3. It is customary to use an average value of 100  $\Omega\text{m}$  if the actual earth resistivity is unknown.

**Table 2-3: Resistivity of different soils [B10]**

| Ground Type      | $\rho$ [ $\Omega\text{m}$ ] |
|------------------|-----------------------------|
| Seawater         | 0.01-1.0                    |
| Wet Organic Soil | 10                          |
| Moist Soil       | 100                         |
| Dry Soil         | 1000                        |
| Bedrock          | $10^4$                      |
| Pure Slate       | $10^7$                      |
| Sandstone        | $10^9$                      |
| Crushed Rock     | $1.5 \times 10^8$           |

All the earth return conductors are the same distance  $D_{kk'}$  from their overhead conductors and have the same resistance  $r_{k'}$ . A circuit representation of the line is shown in Fig. 2.3.



**Fig. 2.3: (a) Complete circuit and (b) simplified circuit representation of series-phase impedances**

Using the circuit in Fig. 2.3 (b) the vector of voltage drops across the conductors is:

$$\begin{bmatrix} V_{Aa} \\ V_{Bb} \\ V_{Cc} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} z_{aa} & z_{ab} & z_{ac} & z_{an1} & z_{an2} \\ z_{ba} & z_{bb} & z_{bc} & z_{bn1} & z_{bn2} \\ z_{ca} & z_{cb} & z_{cc} & z_{cn1} & z_{cn2} \\ \hline z_{n1a} & z_{n1b} & z_{n1c} & z_{n1n1} & z_{n1n2} \\ z_{n2a} & z_{n2b} & z_{n2c} & z_{n2n1} & z_{n2n2} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_{n1} \\ I_{n2} \end{bmatrix} \quad (2-22)$$

The diagonal elements of this matrix are:

$$z_{kk} = r_k + r_{k'} + j\omega 2 \times 10^{-4} \ln \frac{D_{kk'}}{D_{kk}} \quad [\Omega/\text{km}] \quad (2-23)$$

and the off-diagonal elements, for  $k \neq m$ , are:

$$z_{km} = r_k + j\omega 2 \times 10^{-4} \ln \frac{D_{km'}}{D_{km}} \quad [\Omega/\text{km}] \quad (2-24)$$

where

$$k = a, b, c, n1, n2$$

$$m = a, b, c, n1, n2$$

For equations (2-23) and (2-24)  $r_k$  is the AC resistance at 40°C and  $D_{kk}$  the GMR of conductor  $k$ .  $D_{km'}$  is the distance between overhead conductor  $k$  and earth return conductor  $m$ , and  $D_{km}$  the distance between overhead conductor  $k$  and  $m$ .

Next, (2-22) is partitioned as shown above to obtain:

$$\begin{bmatrix} \mathbf{E}_P \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_A & \mathbf{Z}_B \\ \mathbf{Z}_C & \mathbf{Z}_D \end{bmatrix} \begin{bmatrix} \mathbf{I}_P \\ \mathbf{I}_n \end{bmatrix} \quad (2-25)$$

$\mathbf{E}_P$  is the three-dimensional vector of voltage drops across the phase conductors;  $\mathbf{I}_P$  is the three-dimensional vector of phase currents and  $\mathbf{I}_n$  is the two-dimensional vector of neutral currents.

Equation (2-25) is rewritten as two separate matrix equations:

$$\mathbf{E}_P = \mathbf{Z}_A \mathbf{I}_P + \mathbf{Z}_B \mathbf{I}_n \quad (2-26)$$

$$\mathbf{0} = \mathbf{Z}_C \mathbf{I}_P + \mathbf{Z}_D \mathbf{I}_n \quad (2-27)$$

Solving (2-27) for  $\mathbf{I}_n$  and substituting back into (2-26):

$$\mathbf{E}_P = (\mathbf{Z}_A - \mathbf{Z}_B \mathbf{Z}_D^{-1} \mathbf{Z}_C) \mathbf{I}_P \quad (2-28)$$

or

$$\mathbf{E}_P = \mathbf{Z}_P \mathbf{I}_P \quad (2-29)$$

where

$$\mathbf{Z}_P = \mathbf{Z}_A - \mathbf{Z}_B \mathbf{Z}_D^{-1} \mathbf{Z}_C \quad (2-30)$$

$\mathbf{Z}_P$  given by (2-30) is a 3 x 3 series-phase impedance matrix, whose elements are denoted:

$$\mathbf{Z}_P = \begin{bmatrix} Z_{aaeq} & Z_{abeq} & Z_{aceq} \\ Z_{baeq} & Z_{bbeq} & Z_{bceq} \\ Z_{caeq} & Z_{cheq} & Z_{cceq} \end{bmatrix} \quad (2-31)$$



These series impedances are for an untransposed line; if symmetrical component transformations are applied, the results will show mutual coupling between the sequence components [B5], [B8]. Since this is not desirable, the use of symmetrical components is not advised, unless it is assumed that the line is perfectly transposed [A6]. The equations for the transposed line are simply obtained by averaging the diagonal impedances and the off-diagonal impedances:

$$\hat{\mathbf{Z}}_P = \begin{bmatrix} \hat{Z}_S & \hat{Z}_M & \hat{Z}_M \\ \hat{Z}_M & \hat{Z}_S & \hat{Z}_M \\ \hat{Z}_M & \hat{Z}_M & \hat{Z}_S \end{bmatrix} \quad (2-32)$$

where

$$\hat{Z}_S = \frac{1}{3} (Z_{aaeq} + Z_{bbeq} + Z_{ccee}) \quad (2-33)$$

$$\hat{Z}_M = \frac{1}{3} (Z_{abeq} + Z_{bcee} + Z_{caeq}) \quad (2-34)$$

Symmetrical component transformation will give:

$$\hat{\mathbf{Z}}_S = \begin{bmatrix} \hat{Z}_0 & 0 & 0 \\ 0 & \hat{Z}_1 & 0 \\ 0 & 0 & \hat{Z}_2 \end{bmatrix} \quad (2-35)$$

where

$$\hat{Z}_0 = \hat{Z}_S + 2\hat{Z}_M \quad (2-36)$$

$$\hat{Z}_1 = \hat{Z}_2 = \hat{Z}_S - \hat{Z}_M \quad (2-37)$$

These equations give the results that are generally desired for most purposes, but on occasion it is preferable to use the untransposed impedances in terms of the *abc* variables instead of the symmetrical component variables. Of course, in the balanced three-phase case, the positive-sequence resistance is just the resistance of the phase conductor, and the positive sequence reactance is readily calculated from (2-14).

## 2.2.2 Shunt admittance

### Shunt conductance

Conductance accounts for power loss between conductors or between conductors and ground. This power loss is due to leakage currents along insulator strings and corona. In overhead lines its effect is small and usually neglected [B11].

### Shunt capacitance

Firstly, a three-phase, three-wire line with equal phase spacing is considered. The effect of the earth and the neutral conductors is neglected and due to the symmetry only one phase is considered. The positive sequence capacitance, which also equals the negative sequence capacitance, can be calculated as follows:

$$C_1 = C_2 = \frac{2\pi\epsilon}{\ln(D_{eq}/r)} \times 1000 \quad [\text{F/km}] \quad (2-38)$$

where  $r$  is the outside radius of the conductor,  $D_{eq}$  is the *GMD* and  $\epsilon$  is the permittivity of the dielectric medium (for air:  $8.854 \times 10^{-12}$ ). In the equation above  $r$  is assumed to be the outside radius of the conductor, thus treating the stranded conductor as a solid cylindrical conductor. This assumption is valid since most of the charge on a conductor resides on its surface, and thus most of the current will only flow on the surface (skin effect).

To calculate  $C_1 = C_2$  for bundled conductors with bundle spacing  $d$ ,  $r$  is replaced by the modified *GMR* ( $D_{SC}$ ) of the bundle, calculated as follows:

Two-conductor bundle (twin):

$$D_{SC} = \sqrt{rd} \quad (2-39)$$

Three-conductor bundle (triple):

$$D_{SC} = \sqrt[3]{rd^2} \quad (2-40)$$

Four-conductor bundle (quad):

$$D_{SC} = 1.091 \sqrt[4]{rd^3} \quad (2-41)$$

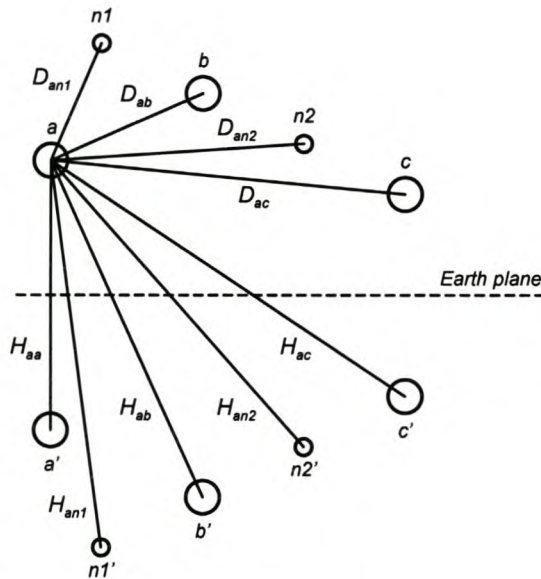
To determine the zero sequence capacitance,  $C_0$ , the effect of neutral conductors and the earth must be considered [B5]. The neutral conductors and the earth also affect



the positive- and negative-sequence capacitance, since they alter the electric field. Equations are subsequently developed for the phase capacitance and sequence capacitance matrices of three-phase lines with neutral conductors.

The earth surface is approximated as a perfectly conducting horizontal plane, even though the earth under the line may have irregular terrain and resistance. The effect of the earth plane is accounted for by the *method of images*, where an image conductor, which has the same radius as the original conductor, replaces the earth. This conductor lies directly below the original conductor and has an equal quantity of negative charge.

Fig. 2.4 shows a three-phase line with neutral conductors and with the earth plane replaced by image conductors.  $D_{km}$  is the distance between overhead conductor  $k$  and  $m$  and  $D_{kk}$  is the outside radius of overhead conductor  $k$  ( $r_k$ ).  $H_{km}$  is the distance between overhead conductor  $k$  and image conductor  $m$ . Since all neutral conductors are grounded to the earth, the voltage between these conductors and the earth will be zero.



**Fig. 2.4: Three-phase line with neutral conductors and with earth plane replaced by image conductors**

In matrix format the voltage between the phase conductors and the earth, and the voltage between the neutral conductors and the earth, are:

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} P_{aa} & P_{ab} & P_{ac} & P_{an1} & P_{an2} \\ P_{ba} & P_{bb} & P_{bc} & P_{bn1} & P_{bn2} \\ P_{ca} & P_{cb} & P_{cc} & P_{cn1} & P_{cn2} \\ P_{n1a} & P_{n1b} & P_{n1c} & P_{n1n1} & P_{n1n2} \\ P_{n2a} & P_{n2b} & P_{n2c} & P_{n2n1} & P_{n2n2} \end{bmatrix} \begin{bmatrix} q_a \\ q_b \\ q_c \\ q_{n1} \\ q_{n2} \end{bmatrix} \quad (2-42)$$

The elements of the 5 x 5  $\mathbf{P}$  matrix are:

$$P_{km} = \frac{1}{2\pi\epsilon} \ln \frac{H_{km}}{D_{km}} \quad [\text{F/m}] \quad (2-43)$$

where

$$k = a, b, c, n1, n2$$

$$m = a, b, c, n1, n2$$

Equation (2-42) is partitioned as shown above to obtain:

$$\begin{bmatrix} \mathbf{V}_P \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_A & \mathbf{P}_B \\ \mathbf{P}_C & \mathbf{P}_D \end{bmatrix} \begin{bmatrix} \mathbf{q}_P \\ \mathbf{q}_n \end{bmatrix} \quad (2-44)$$

$\mathbf{V}_P$  is the three-dimensional vector of phase-to-neutral voltages;  $\mathbf{q}_P$  is the three-dimensional vector of phase-conductor charges and  $\mathbf{q}_n$  is the two-dimensional vector of neutral-conductor charges.

Equation (2-44) is rewritten as two separate equations:

$$\mathbf{V}_P = \mathbf{P}_A \mathbf{q}_P + \mathbf{P}_B \mathbf{q}_n \quad (2-45)$$

$$\mathbf{0} = \mathbf{P}_C \mathbf{q}_P + \mathbf{P}_D \mathbf{q}_n \quad (2-46)$$

Equation (2-46) is solved for  $\mathbf{q}_n$ , which is used in (2-45) to obtain:

$$\mathbf{V}_P = (\mathbf{P}_A - \mathbf{P}_B \mathbf{P}_D^{-1} \mathbf{P}_C) \mathbf{q}_P \quad (2-47)$$

or

$$\mathbf{q}_P = \mathbf{C}_P \mathbf{V}_P \quad (2-48)$$

where

$$\mathbf{C}_P = (\mathbf{P}_A - \mathbf{P}_B \mathbf{P}_D^{-1} \mathbf{P}_C)^{-1} \quad [\text{F/m}] \quad (2-49)$$



$\mathbf{C_P}$  given by (2-49) is the 3 x 3 matrix of phase capacitances whose elements are denoted:

$$\mathbf{C_P} = \begin{bmatrix} C_{aa} & C_{ab} & C_{ac} \\ C_{ba} & C_{bb} & C_{bc} \\ C_{ca} & C_{cb} & C_{cc} \end{bmatrix} \quad [\text{F/m}] \quad (2-50)$$

In general,  $\mathbf{C_P}$  does not satisfy the conditions for a symmetrical capacitance matrix. However, if the line is completely transposed, the diagonal and off-diagonal elements of  $\mathbf{C_P}$  are averaged to obtain:

$$\hat{\mathbf{C_P}} = \begin{bmatrix} \hat{C}_S & \hat{C}_M & \hat{C}_M \\ \hat{C}_M & \hat{C}_S & \hat{C}_M \\ \hat{C}_M & \hat{C}_M & \hat{C}_S \end{bmatrix} \quad [\text{F/m}] \quad (2-51)$$

where

$$\hat{C}_S = \frac{1}{3}(C_{aa} + C_{bb} + C_{cc}) \quad [\text{F/m}] \quad (2-52)$$

$$\hat{C}_M = \frac{1}{3}(C_{ab} + C_{bc} + C_{ca}) \quad [\text{F/m}] \quad (2-53)$$

$\hat{\mathbf{C_P}}$  is a symmetrical matrix whose diagonal terms  $\hat{C}_S$  are positive, and whose off-diagonal terms  $\hat{C}_M$  are negative. This indicates when a positive line-to-neutral voltage is applied to one phase, a positive charge is induced on that phase and negative charge are induced on the other phases, which is physically correct.

The shunt phase admittance matrix is given by:

$$\mathbf{Y_P} = j\omega \mathbf{C_P} = j(2\pi f) \mathbf{C_P} \quad [\text{S/m}] \quad (2-54)$$

or, for a completely transposed line:

$$\hat{\mathbf{Y_P}} = j\omega \hat{\mathbf{C_P}} = j(2\pi f) \hat{\mathbf{C_P}} \quad [\text{S/m}] \quad (2-55)$$

Symmetrical component transformation of (2-55) will give:

$$\hat{\mathbf{Y}}_s = j(2\pi f) \begin{bmatrix} \hat{C}_0 & 0 & 0 \\ 0 & \hat{C}_1 & 0 \\ 0 & 0 & \hat{C}_2 \end{bmatrix} \quad (2-56)$$

where

$$\hat{C}_0 = \hat{C}_s + 2\hat{C}_M \quad [\text{F/m}] \quad (2-57)$$

$$\hat{C}_1 = \hat{C}_2 = \hat{C}_s - \hat{C}_M \quad [\text{F/m}] \quad (2-58)$$

Since  $\hat{C}_M$  is negative, it can be concluded from equations (2-57) and (2-58) that the zero-sequence capacitance,  $\hat{C}_0$ , is usually much less than the positive- or negative-sequence capacitance.

The value calculated by (2-58) for positive- and negative-sequence capacitance will exceed the value given by (2-38) by only about 1% [B4]. Both the effect of the neutral conductors and the earth can therefore be neglected when calculating the positive- and negative-sequence capacitance. In the case of zero-sequence capacitance the neutral conductors may increase the capacitance by an amount of the order of 10%, so that its presence cannot be neglected as may be done for the positive and negative sequences.

### 2.2.3 AC transmission line models for short, medium and long lines

Consider the two-port network in Fig. 2.5, which is a generalised representation of an overhead line. The subscript S denotes sending-end quantities and the subscript R receiving-end quantities.

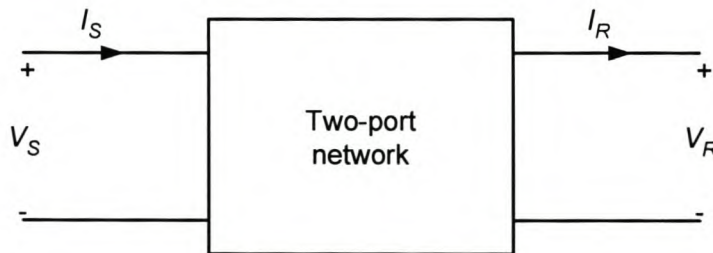


Fig. 2.5: Representation of a two-port network [B5]



The relation between sending- and receiving-end quantities may be expressed in the following general form:

$$V_S = AV_R + BI_R \quad [\text{V}] \quad (2-59)$$

$$I_S = CV_R + DI_R \quad [\text{A}] \quad (2-60)$$

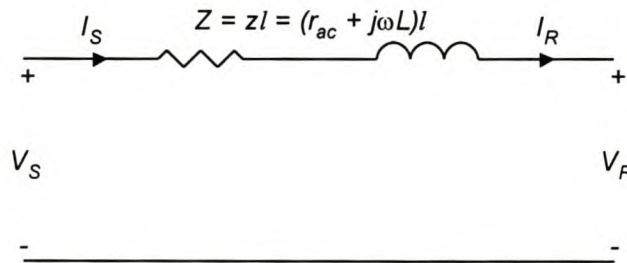
or, in matrix format,

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2-61)$$

The ABCD parameters, sometimes called *generalised circuit constants*, depend on the transmission-line parameters  $R$ ,  $L$ ,  $C$  and  $G$ . The ABCD parameters are, in general, complex numbers.  $A$  and  $D$  are dimensionless.  $B$  and  $C$  have the dimensions of ohms and siemens (mhos), respectively.

### Short and medium line approximations

Fig. 2.6 represents a short line, usually applied to overhead lines less than 80 km long. Only the series resistance and reactance are included, whereas the shunt admittance is neglected. The circuit applies to either single-phase or completely transposed three-phase lines operating under balanced conditions [B5]. For a completely transposed three-phase line,  $Z$  is the positive-sequence series impedance,  $V_S$  and  $V_R$  the positive-sequence line-to-neutral voltages, and  $I_S$  and  $I_R$  the positive-sequence line currents. Similar zero-sequence and negative-sequence circuits can be employed to represent unbalanced operating conditions, using the parameters derived in the previous section.



**Fig. 2.6: Short line [B5]**

From Fig. 2.6 and Fig. 2.7 the following notation is used:

Series impedance per unit length:

$$z = r_{ac} + j\omega L \quad [\Omega/\text{km}] \quad (2-62)$$

Shunt admittance per unit length:

$$y = j\omega C \quad [\text{S/km}] \quad (2-63)$$

Total series impedance:

$$Z = zl \quad [\Omega] \quad (2-64)$$

Total shunt admittance:

$$Y = yl \quad [\text{S}] \quad (2-65)$$

where  $l$  is the length of the line in kilometers. The subscript 1, indicating positive sequence quantities, is omitted in this section.

Writing a KVL and KCL for the short line in Fig. 2.6:

$$V_S = V_R + ZI_R \quad (2-66)$$

$$I_S = I_R \quad (2-67)$$

or in matrix format

$$\begin{bmatrix} V_S \\ I_S \end{bmatrix} = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2-68)$$

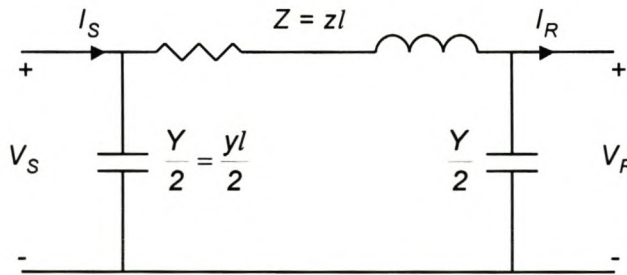
Comparing equations (2-68) and (2-61), the ABCD parameters for the short line are:

$$A = D = 1 \quad [\text{per unit}] \quad (2-69)$$

$$B = Z \quad [\Omega] \quad (2-70)$$

$$C = 0 \quad [\text{S}] \quad (2-71)$$

For medium-length lines, ranging from 80 to 250 km, it is common to lump the total shunt capacitance and locate half at each end of the line [B5]. Such a circuit is called a *nominal  $\pi$  circuit*, shown in Fig. 2.7.



**Fig. 2.7: Medium-length line - nominal  $\pi$  circuit [B5]**



Writing a KVL and KCL for the medium-length line in Fig. 2.7 the following equation is obtained, written in matrix format as:

$$\begin{bmatrix} V_s \\ I_s \end{bmatrix} = \begin{bmatrix} \left(1 + \frac{YZ}{2}\right) & Z \\ Y\left(1 + \frac{YZ}{4}\right) & \left(1 + \frac{YZ}{2}\right) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2-72)$$

Comparing equations (2-72) and (2-61), the ABCD parameters for the medium-length line are:

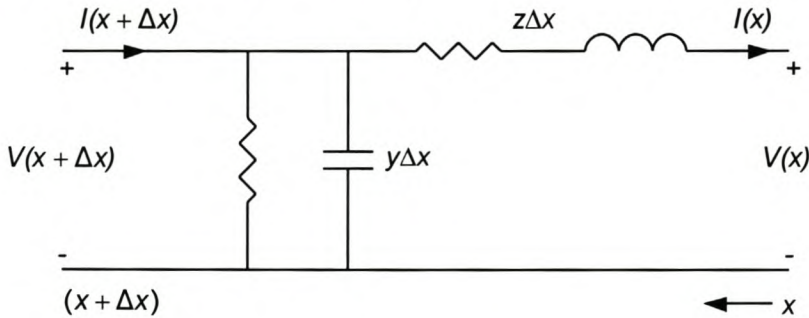
$$A = D = 1 + \frac{YZ}{2} \quad [\text{per unit}] \quad (2-73)$$

$$B = Z \quad [\Omega] \quad (2-74)$$

$$C = Y\left(1 + \frac{YZ}{4}\right) \quad [\text{S}] \quad (2-75)$$

### Overhead-line differential equations

Long overhead-line equations (in terms of the ABCD parameters) will be discussed, as these represent an exact model for the overhead line. Fig. 2.1 represents a line section of length  $\Delta x$  and are repeated in Fig. 2.8 for convenience.



**Fig. 2.8: Overhead line section of length  $\Delta x$  [B5]**

The position  $x$  is measured from the receiving end of the line. The circuit constants:

$$z = r_{ac} + j\omega L \quad [\Omega/\text{km}] \quad (2-76)$$

$$y = G + j\omega C \quad [\text{S}/\text{km}] \quad (2-77)$$

are distributed uniformly throughout the length of the line.  $G$  is usually neglected for overhead lines [B7].

Equation (2-78) gives the ABCD parameters of the distributed line as a function of distance:

$$\begin{bmatrix} V(x) \\ I(x) \end{bmatrix} = \begin{bmatrix} A(x) & B(x) \\ C(x) & D(x) \end{bmatrix} \begin{bmatrix} V_R \\ I_R \end{bmatrix} \quad (2-78)$$

where

$$A(x) = D(x) = \cosh(\gamma x) \quad [\text{per unit}] \quad (2-79)$$

$$B(x) = Z_c \sinh(\gamma x) \quad [\Omega] \quad (2-80)$$

$$C(x) = \frac{1}{Z_c} \sinh(\gamma x) \quad [\text{S}] \quad (2-81)$$

$Z_c = \sqrt{z/y} \quad [\Omega]$  is called the *characteristic impedance* of the line, and  $\gamma = \sqrt{zy} \text{ km}^{-1}$  the *propagation constant*.

Equation (2-78) gives the voltage and current anywhere along the line, at a distance  $x$  from the receiving end. Letting  $x = l$ , the length of the line in kilometres, we obtain the voltage and current at the sending end of the line:

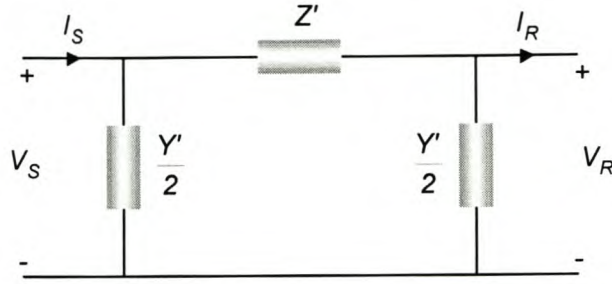
$$V_S = V_R \cosh(\gamma l) + I_R Z_c \sinh(\gamma l) \quad (2-82)$$

$$I_S = \frac{V_R}{Z_c} \sinh(\gamma l) + I_R \cosh(\gamma l) \quad (2-83)$$

The ABCD parameters given by equations (2-79) to (2-81) are exact parameters valid for any line length. For accurate calculations these equations must be used for overhead lines longer than 250 km [B5]. For quick calculations by hand, the approximate parameters discussed in the previous sections for short or medium lines can be used. For more accurate calculations the exact ABCD parameters must be used as well for short and medium lines.

Many computer programs, used in power system analysis and design, assume a circuit representation of an overhead line. The *overhead line equivalent  $\pi$  circuit*, shown in Fig. 2.9, are therefore a more convenient way of representing the terminal characteristics of an overhead line than the ABCD parameters of equations (2-79) to (2-81).





**Fig. 2.9: Overhead line equivalent  $\pi$  circuit [B5]**

From Fig. 2.9:

$$Z' = Z_c \sinh(\gamma l) \quad (2-84)$$

$$\frac{Y'}{2} = \frac{\tanh(\gamma l/2)}{Z_c} \quad (2-85)$$

Fig. 2.9 is identical in structure to the nominal  $\pi$  circuit of Fig. 2.7, except that  $Z'$  and  $Y'$  are used instead of  $Z$  and  $Y$ .  $Z'$  and  $Y'$  (equations (2-84) and (2-85)) are chosen in such a way that the equivalent  $\pi$  circuit will have the same ABCD parameters as those of the distributed line (equations (2-79) to (2-81)). Table 2-4 summarises the ABCD parameters for short, medium and long overhead lines.

**Table 2-4: Summary of Overhead Line Parameters [B5]**

| Parameter   | $A = D$<br>[per unit]                  | $B$ [ $\Omega$ ]           | $C$ [S]   |
|---|--|----------------------------|---|
| Short line (less than 80 km)                            | 1                                      | $Z$                        | 0   |
| Medium line – nominal $\pi$ circuit (80 to 250 km)      | $1 + \frac{YZ}{2}$                     | $Z$                        | $Y \left( 1 + \frac{YZ}{4} \right)$                                   |
| Long line – equivalent $\pi$ circuit (more than 250 km) | $\cosh(\gamma l) = 1 + \frac{Y'Z'}{2}$ | $Z_c \sinh(\gamma l) = Z'$ | $(1/Z_c) \sinh(\gamma l)$<br>$= Y' \left( 1 + \frac{Y'Z'}{4} \right)$ |

## 2.2.4 Implementation

This section focuses on the development of the Overhead Line Parameters (OLP) software tool, which was written as an extension to the input interface of PSAT (Power System Analysis Tool) to automatically calculate the equivalent impedances of a transmission line. OLP is restricted to calculating the series impedance and shunt

admittance of balanced three-phase, three-wire lines, where it is assumed that the lines are perfectly transposed. Because currently only three-phase, balanced circuits are considered in PSAT, these restrictions are acceptable.

An ACSR conductor and tower database is implemented in OLP. Input data consist of selecting a phase conductor, a tower configuration and the number of phase conductors in a bundle. The values for bundle spacing, earth resistivity and distance of the line must also be entered. Output data consist of the series impedance and shunt admittance of different ACSR conductor sizes, bundle selections and tower configurations. The equivalent  $\pi$  circuit discussed in section 2.2.3 is also calculated. Provision is already made for unbalanced conditions as both the positive- and zero-sequence values are calculated. The results can be exported to either the sending or the receiving end in the input interface of PSAT [A1], [C2].

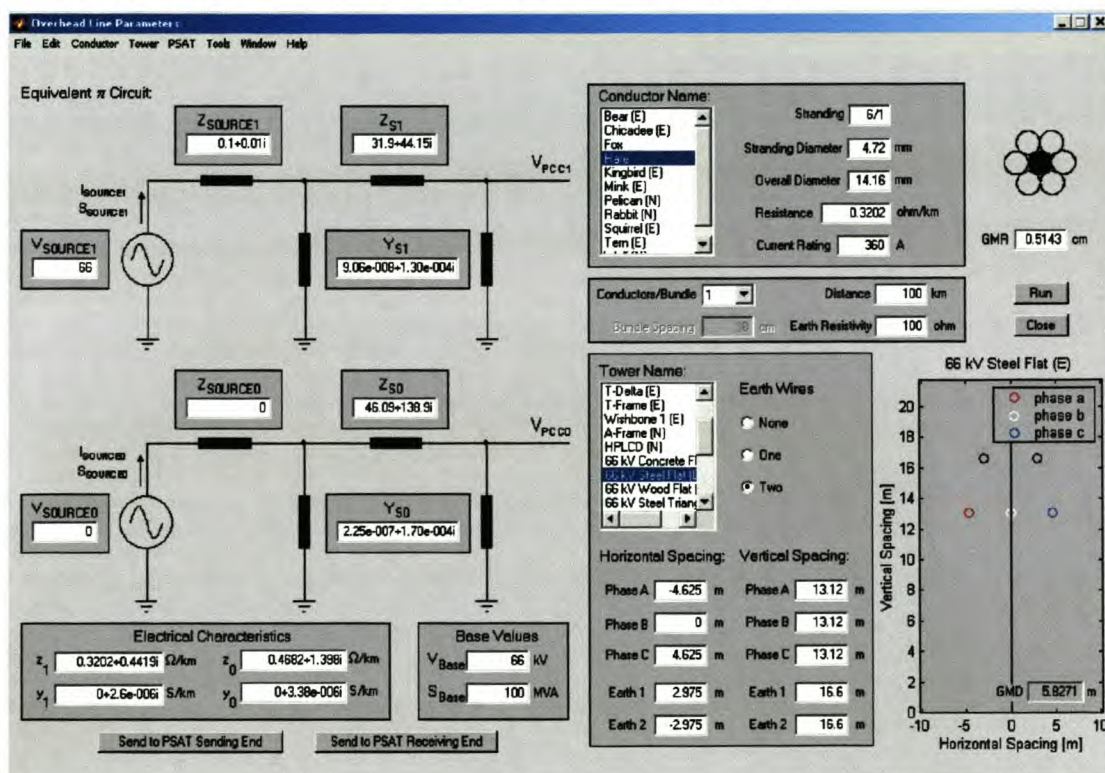


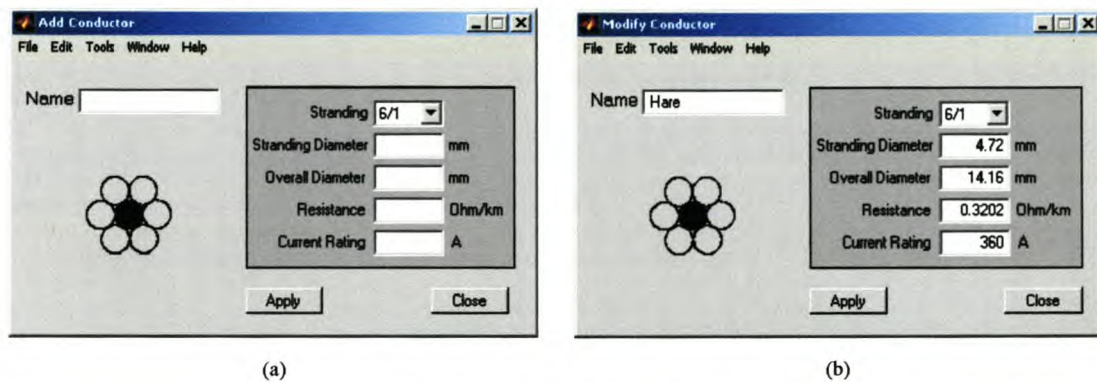
Fig. 2.10: Overhead line parameters interface

The OLP interface is shown in Fig. 2.10. Different conductors are selected from the conductor database by clicking on the conductor name in the *Conductor Name* listbox. Similarly, different towers can be selected from the tower database by clicking on the tower name in the *Tower Name* listbox. The tower structure is graphically portrayed as a single black line in the center: the red, blue and white



circles represent the three phases and the black circles the neutral wires. The calculated sequence values for the series impedance and shunt admittance are displayed in the *Electrical Characteristics* box and are used to calculate the equivalent  $\pi$  circuit.

To add new conductors to the conductor database or modify existing conductors in the database, *Add Conductor* or *Modify Conductor* can be selected from the *Conductor* menu in the OLP interface. The add conductor and modify conductor interface of OLP is shown in Fig. 2.11.



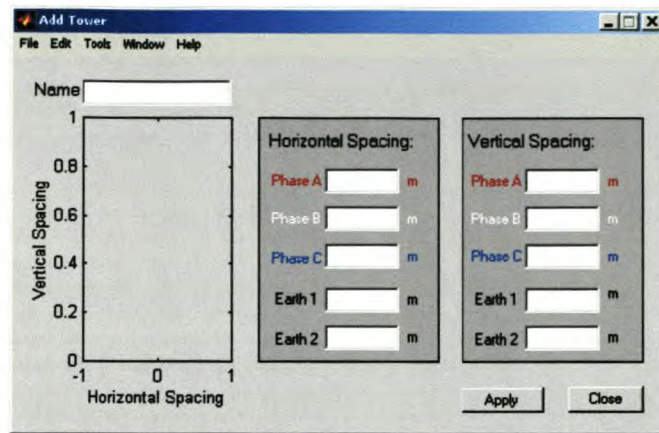
**Fig. 2.11: a) Add conductor and b) Modify conductor interface**

Similarly, new towers can be added to the tower database or existing towers can be modified by selecting *Add Tower* or *Modify Tower* from the *Tower* menu in the OLP interface. These interfaces are shown in Fig. 2.12.

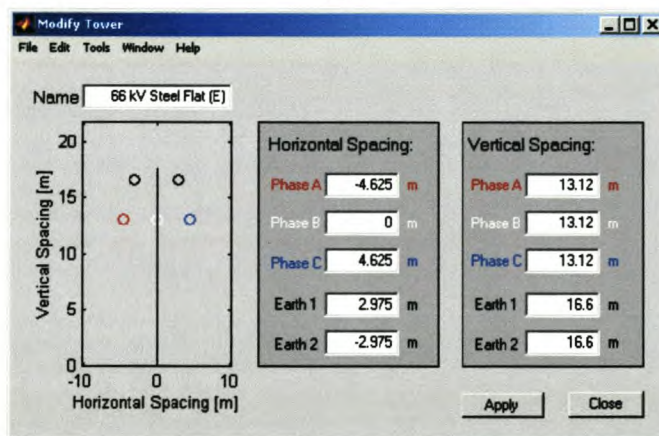
OLP is used to quickly calculate the series impedance, shunt admittance and equivalent  $\pi$  circuit for different ACSR conductor sizes, bundle selections and tower configurations. Calculation time and effort are therefore greatly reduced and the results can be exported to PSAT.

## Software development

OLP is implemented using Matlab, developed by Mathworks Inc. Matlab was chosen as it is fundamentally a mathematical package, but it also supports the development of object-orientated graphical user interfaces (GUIs). Equations and formulae implemented in Matlab can easily be exported to other simulation packages that allow the user to create custom devices. It also provides excellent graphing tools that are not always available in other simulation packages.



(a)



(b)

**Fig. 2.12: a) Add tower and b) Modify tower interface**

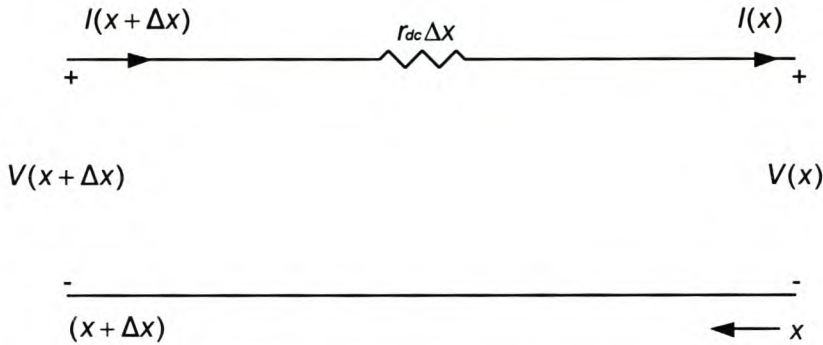
Matlab GUIs are built around figures. A function is linked to each object in the figure and is executed according to the selections made or inputs given by the user. Matlab provides excellent printed and on-line documentation regarding GUIs [B12], and therefore the functions pertaining to the GUI and other graphics are not discussed here.

The equations to calculate the series impedance, shunt admittance and equivalent  $\pi$  circuit for a specific tower structure with different ACSR conductors, bundle selections, distances and earth resistivity are implemented in *conductor.m*. The equations to calculate the series impedance, shunt admittance and equivalent  $\pi$  circuit for different tower configurations are implemented in *tower.m*. The functions to add new conductors and modify existing conductors or to add new towers and modify existing towers are included in *conductor.m* and *tower.m* respectively. The development of the equations was discussed in sections 2.2.1 to 2.2.3. The full program listing for *conductor.m* and *tower.m* is included in Addendum B.1.



## 2.3 THE DC TRANSMISSION LINE MODEL

Fig. 2.13 represents a line section of length  $\Delta x$ . Whereas the configuration of AC lines is determined by the number of phases, the equivalent for DC lines are the poles. Since there is no inductance, there will not be any stability problems. There will be no line-charging currents, because there is no capacitance. Therefore, there will be no reactive power injected into or absorbed by the power system and only real power will flow.



**Fig. 2.13: DC line section of length  $\Delta x$**

The DC resistance of a conductor at a specified temperature  $T$  is:

$$r_{dc,T} = \frac{\rho_T l}{A} \quad [\Omega] \quad (2-86)$$

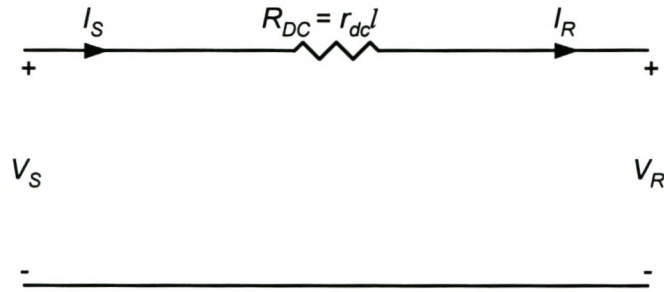
where  $\rho_T$  = conductor resistivity at temperature  $T$

$l$  = conductor length

$A$  = conductor cross-sectional area.

For stranded conductors alternate layers of strands are spiralled in opposite directions to hold the strands together. This makes the strands 1 or 2% longer than the actual conductor length. As a result of this, the DC resistance of a stranded conductor is 1 or 2% larger than that calculated from equation (2-86) for a specified conductor length. It is very seldom that resistance is calculated from first principles, since the DC resistance values are readily available from the conductor tables at 20°C. These values also include the spiralling effect for stranded conductors. Equation (2-2) is used to correct  $r_{dc}$  to the required operating temperature as was discussed in detail in section 2.2.1.

For bundled conductors, the effective resistance per unit length of the bundled conductor is equal to  $r_{dc}/n$ , where  $n$  is the amount of conductors per bundle ( $n = 2, 3$ , or 4) [B5].



**Fig. 2.14: DC transmission line model**

The DC transmission line model is illustrated in Fig. 2.14. Writing a KVL and KCL for the line:

$$V_S = V_R + R_{DC} I_{DC} \quad (2-87)$$

$$I_S = I_R = I_{DC} \quad (2-88)$$

Comparing equations (2-167), (2-168) and (2-61) the ABCD parameters for the DC line are:

$$A = D = 1 \quad [\text{per unit}] \quad (2-89)$$

$$B = R_{DC} \quad [\Omega] \quad (2-90)$$

$$C = 0 \quad [\text{S}] \quad (2-91)$$



## 2.4 VOLTAGE REGULATION AND LINE LOSSES FOR AC TRANSMISSION LINES

Voltage regulation and efficiency (percentage losses) are fundamental properties of the performance of a transmission line. Losses depend on the impedance of a line. For AC transmission, line impedance is a function of the phasing (three-phase, phase-phase, etc.), conductor AC resistance and conductor spacing [C6].

Line losses can be reduced by either increasing the conductor size or by increasing the supply voltage for constant power loads or lowering it for constant impedance loads. Where the load is of constant power type, like most industrial loads, increasing the supply voltage is effective. The drawback of increasing the voltage is that the equipment is stressed, which in turn reduces the lifespan. This increases the probability of flashovers and failures, which in turn reduce the reliability of supply [C6]. Therefore losses cannot be reduced through acceptable means other than increasing the conductor size, which is a very costly option.

### 2.4.1 Power flow in terms of ABCD constants

Although power flow at any point along an overhead line can be found if the voltage, current and power factor at that point are known or can be calculated, equations for power can also be derived in terms of ABCD constants. Also, many times the only known variables in the system are the sending-end voltage at the source bus (also called the slack or swing bus), the line length, the line parameters and the receiving-end power.

The two-port network of Fig. 2.5 is repeated in Fig. 2.15 for convenience, as well as equations (2-59) and (2-60).

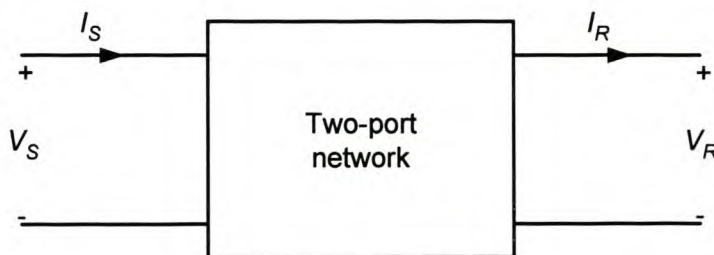


Fig. 2.15: Representation of a two-port network [B5]

From equations (2-59) and (2-60):

$$V_S = AV_R + BI_R \quad [\text{V}] \quad (2-92)$$

$$I_S = CV_R + DI_R \quad [\text{A}] \quad (2-93)$$

The following notation is used:

$$A = |A| \angle \alpha \quad (2-94)$$

$$B = |B| \angle \beta \quad (2-95)$$

$$V_R = |V_R| \angle 0^\circ \quad (2-96)$$

$$V_S = |V_S| \angle \delta \quad (2-97)$$

The ABCD parameters are derived for line-to-neutral voltages; therefore  $V_S$  and  $V_R$  must also be treated as such.

Substituting (2-94) and (2-95) into (2-92) and solving for  $I_R$ :

$$I_R = \frac{1}{|B|} (|V_S| \angle (\delta - \beta) - |A| |V_R| \angle (\alpha - \beta)). \quad (2-98)$$

Then, the complex power per phase  $V_R I_R^*$  at the receiving end is:

$$S_R = P_R + jQ_R = \frac{|V_R|}{|B|} (|V_S| \angle (\beta - \delta) - |A| |V_R| \angle (\beta - \alpha)) \quad (2-99)$$

where  $I_R^*$  is the complex conjugate of  $I_R$ .

The real power ( $P_R$ ) and reactive power ( $Q_R$ ) delivered to the receiving end are:

$$P_R = \text{Re}(S_R) = \frac{|V_R| |V_S|}{|B|} \cos(\beta - \delta) - \frac{|A| |V_R|^2}{|B|} \cos(\beta - \alpha) \quad (2-100)$$

$$Q_R = \text{Im}(S_R) = \frac{|V_R| |V_S|}{|B|} \sin(\beta - \delta) - \frac{|A| |V_R|^2}{|B|} \sin(\beta - \alpha) \quad (2-101)$$

## 2.4.2 Calculation of the sending-end voltage

The content of this section and sections 2.4.3 to 2.4.4 is based on work done by Smit [C7].



Only the magnitude  $|V_S|$  of the sending-end voltage is usually known, where the angle  $\delta$  is actually unknown and must be eliminated from equations (2-100) and (2-101). The next few steps indicate how to calculate  $V_S$  from the basic information available.

Solving (2-100) for  $\cos(\beta - \delta)$ :

$$\cos(\beta - \delta) = \frac{|B|}{|V_R||V_S|} \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right) = K_1 \quad (2-102)$$

and similarly (2-101) for  $\sin(\beta - \delta)$ :

$$\sin(\beta - \delta) = \frac{|B|}{|V_R||V_S|} \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right) = K_2 \quad (2-103)$$

The angle  $\delta$  is unknown and is eliminated from (2-102) and (2-103) using the following identity:

$$\cos^2(\beta - \delta) + \sin^2(\beta - \delta) = 1 \Rightarrow K_1^2 + K_2^2 = 1 \quad (2-104)$$

Substituting equations (2-102) and (2-103) into equation (2-104):

$$\begin{aligned} \left( \frac{|B|}{|V_R||V_S|} \right)^2 \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right)^2 + \\ \left( \frac{|B|}{|V_R||V_S|} \right)^2 \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right)^2 = 1 \end{aligned} \quad (2-105)$$

Solving for  $V_S$  from (2-105) in terms of  $V_R$ ,  $P_R$  and  $Q_R$ :

$$|V_S|^2 = \left( \frac{|B|}{|V_R|} \right)^2 \left[ \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right)^2 + \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right)^2 \right] \quad (2-106)$$

From equation (2-106):

$$\begin{aligned} \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right)^2 &= P_R^2 + 2P_R \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \\ &+ \frac{|A|^2|V_R|^4}{|B|^2} \cos^2(\beta - \alpha) \end{aligned} \quad (2-107)$$

and

$$\begin{aligned} \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right)^2 &= Q_R^2 + 2Q_R \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \\ &+ \frac{|A|^2|V_R|^4}{|B|^2} \sin^2(\beta - \alpha) \end{aligned} \quad (2-108)$$

With  $S_R^2 = P_R^2 + Q_R^2$  and using the same identity as in (2-104), (2-106) can be simplified to:

$$|V_S|^2 = \frac{|B|^2 S_R^2}{|V_R|^2} + |A|^2 |V_R|^2 + 2|A||B|[P_R \cos(\beta - \alpha) + Q_R \sin(\beta - \alpha)] \quad (2-109)$$

The angles  $\alpha$  and  $\beta$  are network dependent and are eliminated by replacing the angles with the appropriate values of the equivalent  $\pi$  model of the line. It is also preferable to work with the real variables available, as they are easier to implement in a program or spreadsheet.

Using the equivalent  $\pi$  model of Fig. 2.9 and equations (2-84) and (2-85), the following formulae apply:

$$Z' = Z_C \sinh(\delta l) = R' + jX' \quad (2-110)$$

$$\frac{Y'}{2} = \frac{\tanh(\gamma l)}{Z_C} = \frac{G'}{2} + j \frac{B'}{2} \quad (2-111)$$

The shunt conductance  $G'$ , introduced in (2-111), is small and its effect is often neglected [B5], [B11].

From Table 2-4 the following:



$$\begin{aligned}
 A &= |A| \angle \alpha = 1 + \frac{Y' Z'}{2} = D = |D| \angle \Delta \\
 \Rightarrow A &= 1 + (R' + jX') \left( \frac{jB'}{2} \right) \\
 \Rightarrow A &= \left( 1 - \frac{B' X'}{2} \right) + j \frac{R' B'}{2}
 \end{aligned} \tag{2-112}$$

and

$$\begin{aligned}
 B &= |B| \angle \beta = Z' \\
 \Rightarrow B &= R' + jX'
 \end{aligned} \tag{2-113}$$

In the equations to follow the subscript (') indicating equivalent  $\pi$  circuit values will be neglected for the sake of simplicity and to avoid confusion. From equations (2-112) and (2-113) it further follows that:

$$\cos \alpha = \cos \Delta = \frac{\left( 1 - \frac{BX}{2} \right)}{|D|} \tag{2-114}$$

$$\sin \alpha = \sin \Delta = \frac{RB}{2|D|} \tag{2-115}$$

$$\cos \beta = \frac{R}{|Z|} \tag{2-116}$$

and

$$\sin \beta = \frac{X}{|Z|} \tag{2-117}$$

Substituting the values for  $|A|$  and  $|B|$  into equation (2-109) and remembering that  $|B|$  is one of the transmission line parameters equal to  $Z'$  with the subscript neglected:

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2 |V_R|^2 + 2|D||Z|[P_R \cos(\beta - \alpha) + Q_R \sin(\beta - \alpha)] \tag{2-118}$$

Using the following identities:

$$\cos(\beta - \alpha) = \cos \beta \cos \alpha + \sin \beta \sin \alpha \tag{2-119}$$

$$\sin(\beta - \alpha) = \sin \beta \cos \alpha - \cos \beta \sin \alpha \tag{2-120}$$

equation (2-118) can now be rewritten as:

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2 |V_R|^2 + 2|D||Z| \left[ P_R (\cos \beta \cos \alpha + \sin \beta \sin \alpha) + Q_R (\sin \beta \cos \alpha - \cos \beta \sin \alpha) \right] \quad (2-121)$$

By further substitution of equations (2-114) to (2-117) into equation (2-121):

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + |D|^2 |V_R|^2 + 2|D||Z| \left[ P_R \left( \frac{R}{|Z|} \frac{\left(1 - \frac{BX}{2}\right)}{|D|} + \frac{X}{|Z|} \frac{RB}{2|D|} \right) + Q_R \left( \frac{X}{|Z|} \frac{\left(1 - \frac{BX}{2}\right)}{|D|} - \frac{R}{|Z|} \frac{RB}{2|D|} \right) \right] \quad (2-122)$$

Substituting  $|D|^2$  in equation (2-122) with:

$$|D|^2 = \left(1 - \frac{BX}{2}\right)^2 + \left(\frac{RB}{2}\right)^2 \quad (2-123)$$

and simplifying

$$|V_S|^2 = \frac{|Z|^2 S_R^2}{|V_R|^2} + \left[1 - BX + \left(\frac{B|Z|}{2}\right)^2\right] |V_R|^2 + 2RP_R + Q_R(2X - B|Z|^2) \quad (2-124)$$

where

$$|Z|^2 = R^2 + X^2 \quad (2-125)$$

### 2.4.3 Calculation of the receiving-end voltage

If  $|V_S|$  is known  $|V_R|$  can be calculated by evaluating equation (2-124) with the general expression for a quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2-126)$$

where  $x$  is the unknown quantity and  $a$ ,  $b$  and  $c$  are numerical factors referred to as coefficients of the equation.

Equation (2-124) can be rewritten as:

$$|V_S|^2 - 2RP_R - Q_R(2X - B|Z|^2) = \frac{|Z|^2 S_R^2}{|V_R|^2} + \left[1 - BX + \left(\frac{B|Z|}{2}\right)^2\right] |V_R|^2 \quad (2-127)$$



Equation (2-127) is reduced to:

$$F = \frac{|Z|^2 S_R^2}{|V_R|^2} + K|V_R|^2 \quad (2-128)$$

where

$$F = |V_S|^2 - 2RP_R - Q_R(2X - B|Z|^2) \quad (2-129)$$

and

$$K = 1 - BX + \frac{B^2|Z|^2}{4} \quad (2-130)$$

Equation (2-128) is now written in the form of (2-126) as:

$$|V_R| = \sqrt{\frac{F + \sqrt{F^2 - 4|Z|^2 S_R^2 K}}{2K}} \quad (2-131)$$

The receiving-end voltage ( $|V_R|$ ) can now be determined by calculating the values for  $F$  and  $K$  in (2-129) and (2-130) and substituting these values into (2-131).

After  $|V_R|$  is determined from equation (2-131), the *voltage regulation* of the line can be expressed as:

$$V_d = (V_S - V_R) \times 100\% \quad (2-132)$$

where  $V_S$  and  $V_R$  are per unit quantities.

#### 2.4.4 Calculation of the sending-end power

In order to calculate the line losses the sending-end power needs to be calculated. The complex power at the sending end of the line is given by:

$$S_S = V_S I_S^* \quad (2-133)$$

From section 2.4.1 the sending-end voltage and current was expressed in terms of the ABCD parameters as:

$$V_S = AV_R + BI_R \quad [\text{V}] \quad (2-134)$$

$$I_S = CV_R + DI_R \quad [\text{A}] \quad (2-135)$$

$I_R$  is the only unknown variable in (2-135) and can be solved for by substituting (2-134) into (2-135) and simplifying:

$$\begin{aligned} I_S &= CV_R + \frac{D}{B}(V_S - AV_R) \\ \Rightarrow I_S &= \frac{D}{B}V_S - \frac{V_R}{B}(AD - BC) \\ \Rightarrow I_S &= \frac{1}{B}(DV_S - V_R) \end{aligned} \quad (2-136)$$

where  $AD - BC = 1$ .

The complex conjugate of  $I_S$  will thus be:

$$I_S^* = \frac{1}{|B|}(|D||V_S|\angle(\beta - \Delta - \delta) - |V_R|\angle\beta) \quad (2-137)$$

The complex power at the sending end is calculated by substituting (2-137) into (2-133) and simplifying:

$$S_S = P_S + jQ_S = \frac{|V_S|}{|B|}(|D||V_S|\angle(\beta - \Delta) - |V_R|\angle(\beta + \delta)) \quad (2-138)$$

The real power ( $P_S$ ) and reactive power ( $Q_S$ ) at the sending end are:

$$P_S = \frac{|D||V_S|^2}{|B|}\cos(\beta - \Delta) - \frac{|V_S||V_R|}{|B|}\cos(\beta + \delta) \quad (2-139)$$

$$Q_S = \frac{|D||V_S|^2}{|B|}\sin(\beta - \Delta) - \frac{|V_S||V_R|}{|B|}\sin(\beta + \delta). \quad (2-140)$$

The angle  $\delta$  in equations (2-139) and (2-140) are unknown and must be eliminated, using equations (2-102) and (2-103), which are repeated in (2-141) and (2-142) for convenience.

$$\cos(\beta - \delta) = K_1 \quad (2-141)$$

$$\sin(\beta - \delta) = K_2 \quad (2-142)$$

Using trigonometric identities (2-141) and (2-142) can be rewritten as:



$$\begin{aligned}\cos(\beta - \delta) &= \cos \beta \cos \delta + \sin \beta \sin \delta = K_1 \\ \Rightarrow \cos(\delta) &= \frac{(K_1 - \sin \beta \sin \delta)}{\cos \beta}\end{aligned}\quad (2-143)$$

and

$$\begin{aligned}\sin(\beta - \delta) &= \sin \beta \cos \delta - \cos \beta \sin \delta = K_2 \\ \Rightarrow \sin \delta &= \frac{(\sin \beta \cos \delta - K_2)}{\cos \beta}\end{aligned}\quad (2-144)$$

Substitute (2-143) into (2-144) and simplify:

$$\begin{aligned}\sin \delta &= \frac{\left( \sin \beta \cdot \frac{(K_1 - \sin \beta \cdot \sin \delta)}{\cos \beta} - K_2 \right)}{\cos \beta} \\ \Rightarrow \sin \delta \left( 1 + \frac{\sin^2 \beta}{\cos^2 \beta} \right) &= \frac{\sin \beta}{\cos^2 \beta} K_1 - \frac{K_2}{\cos \beta} \\ \Rightarrow \sin \delta &= \sin(\beta) K_1 - \cos(\beta) K_2\end{aligned}\quad (2-145)$$

Equation (2-145) is now substituted back into (2-143) and simplified:

$$\begin{aligned}\cos \delta &= \frac{(K_1 - \sin \beta [\sin(\beta) K_1 - \cos(\beta) K_2])}{\cos \beta} \\ \Rightarrow \cos \delta &= \frac{K_1}{\cos \beta} (1 - \sin^2 \beta) + \sin(\beta) K_2 \\ \Rightarrow \cos \delta &= \cos(\beta) K_1 + \sin(\beta) K_2\end{aligned}\quad (2-146)$$

Expanding equation (2-139):

$$P_s = \frac{|D||V_s|^2}{|B|} \cos(\beta - \Delta) - \frac{|V_s||V_R|}{|B|} (\cos \beta \cos \delta - \sin \beta \sin \delta) \quad (2-147)$$

The term  $(\cos \beta \cos \delta - \sin \beta \sin \delta)$  can now be rewritten as:

$$\begin{aligned}\cos \beta \cos \delta - \sin \beta \sin \delta &= K_1 (\cos^2 \beta - \sin^2 \beta) + K_2 (\cos \beta \sin \beta + \cos \beta \sin \beta) \\ \Rightarrow K &= K_1 (\cos^2 \beta - \sin^2 \beta) + 2K_2 \cos \beta \sin \beta\end{aligned}\quad (2-148)$$

The values of  $K_1$  and  $K_2$  (equations (2-102) and (2-103)) are now substituted into (2-148):

$$K = \frac{|B|(\cos^2 \beta - \sin^2 \beta)}{|V_R||V_S|} \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right) + \frac{2 \cos \beta \sin \beta |B|}{|V_R||V_S|} \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right) \quad (2-149)$$

Substitute (2-149) into (2-147) and simplify:

$$P_S = \frac{|D||V_S|^2}{|B|} \cos(\beta - \Delta) - (\cos^2 \beta - \sin^2 \beta) \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right) - 2 \cos \beta \sin \beta \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right) \quad (2-150)$$

$$\Rightarrow P_S = \frac{|D||V_S|^2}{|B|} \cos(\beta - \Delta) - M - N$$

where

$$M = (\cos^2 \beta - \sin^2 \beta) \left( P_R + \frac{|A||V_R|^2}{|B|} \cos(\beta - \alpha) \right) \quad (2-151)$$

and

$$N = 2 \cos \beta \sin \beta \left( Q_R + \frac{|A||V_R|^2}{|B|} \sin(\beta - \alpha) \right). \quad (2-152)$$

The angles  $\alpha$ ,  $\beta$ , and  $\Delta$  are network dependent and are replaced by the appropriate values of the equivalent  $\pi$  circuit model. From section 2.4.2 the following:

$$\cos \alpha = \cos \Delta = \frac{\left(1 - \frac{BX}{2}\right)}{|D|} \quad (2-153)$$

$$\sin \alpha = \sin \Delta = \frac{RB}{2|D|} \quad (2-154)$$

$$\cos \beta = \frac{R}{|Z|} \quad (2-155)$$

$$\sin \beta = \frac{X}{|Z|} \quad (2-156)$$



Expanding the sine and cosine terms in equations (2-150) - (2-152) and substituting equations (2-153) - (2-156):

$$P_s = \frac{|D||V_s|^2}{|Z|} \left( \frac{R}{|Z|} \frac{\left(1 - \frac{BX}{2}\right)}{|D|} + \frac{X}{|Z|} \frac{RB}{2|D|} \right) - M - N \quad (2-157)$$

with

$$M = \left( \frac{R^2 - X^2}{|Z|^2} \right) \left[ P_R + \frac{|A||V_R|^2}{|Z|} \left( \frac{R}{|Z|} \frac{\left(1 - \frac{BX}{2}\right)}{|D|} + \frac{X}{|Z|} \frac{RB}{2|D|} \right) \right] \quad (2-158)$$

and

$$N = 2 \frac{RX}{|Z|^2} \left[ Q_R + \frac{|A||V_R|^2}{|Z|} \left( \frac{X}{|Z|} \frac{\left(1 - \frac{BX}{2}\right)}{|D|} + \frac{R}{|Z|} \frac{RB}{2|D|} \right) \right]. \quad (2-159)$$

Substitute (2-158) and (2-159) into (2-157) and simplify:

$$P_s = \frac{|V_s|^2 R - P_R (R^2 - X^2) - 2RXQ_R}{|Z|^2} - \frac{|V_R|^2}{|Z|^4} \left( R(R^2 - X^2) + 2RX \left( X - \frac{B}{2} |Z|^2 \right) \right) \quad (2-160)$$

Some of the terms in (2-160) are now reduced as follows:

$$\begin{aligned} (R^2 - X^2) &= R^2 - X^2 + 2R^2 - 2R^2 \\ \Rightarrow (R^2 - X^2) &= 2R^2 - |Z|^2 \end{aligned} \quad (2-161)$$

and

$$R(R^2 - X^2) + 2RX \left( X - \frac{B}{2} |Z|^2 \right) = R|Z|^2 (1 - BX). \quad (2-162)$$

Substituting (2-161) and (2-162) into (2-160) and simplifying:

$$P_s = \frac{R}{|Z|^2} \left( (|V_s|^2 - |V_R|^2) - 2(RP_R + XQ_R) + BX|V_R|^2 \right) + P_R \quad (2-163)$$

The real line losses can be defined as the difference between the sending-end real power  $P_S$  and the receiving-end real power  $P_R$ :

$$P_{line\ loss} = \frac{R}{|Z|^2} \left( |V_S|^2 - |V_R|^2 \right) - 2(RP_R + XQ_R) + BX|V_R|^2 \quad (2-164)$$

The receiving-end reactive power  $Q_R$  can be calculated in the same way as the real power  $P_R$ :

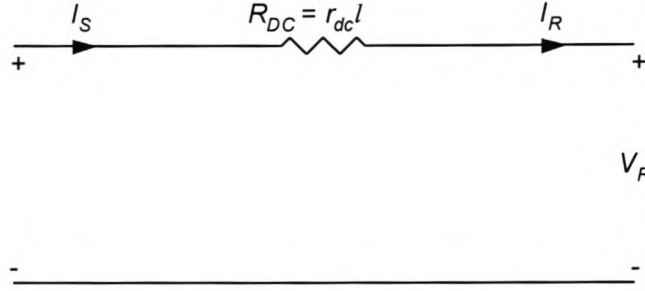
$$Q_S = \frac{X}{|Z|^2} \left( |V_S|^2 - |V_R|^2 - 2(RP_R + XQ_R) \right) - \frac{B}{|Z|^2} \left( \frac{|Z|^2}{2} \left( |V_S|^2 - |V_R|^2 \right) + R^2|V_R|^2 \right) \quad (2-165)$$

Similarly the reactive line losses can be defined as the difference between the sending-end reactive power  $Q_S$  and the receiving-end reactive power  $Q_R$ :

$$Q_{line\ loss} = \frac{X}{|Z|^2} \left( |V_S|^2 - |V_R|^2 - 2(RP_R + XQ_R) \right) - \frac{B}{|Z|^2} \left( \frac{|Z|^2}{2} \left( |V_S|^2 - |V_R|^2 \right) + R^2|V_R|^2 \right) \quad (2-166)$$



## 2.5 VOLTAGE REGULATION AND LINE LOSSES FOR DC TRANSMISSION LINES



**Fig. 2.16: DC transmission line model**

The DC transmission line model is illustrated in Fig. 2.14 and is repeated in Fig. 2.16 for the purposes of this discussion. Writing a KVL and KCL for the line:

$$V_S = V_R + R_{DC} I_{DC} \quad [\text{V}] \quad (2-167)$$

$$I_S = I_R = I_{DC} \quad [\text{A}] \quad (2-168)$$

Comparing equations (2-167), (2-168) and (2-61) the ABCD parameters for the DC line are:

$$A = D = 1 \quad [\text{per unit}] \quad (2-169)$$

$$B = R_{DC} \quad [\Omega] \quad (2-170)$$

$$C = 0 \quad [\text{S}] \quad (2-171)$$

In the case of DC transmission line inductance and capacitance have no effect and equations (2-169) to (2-171) will be treated as real values, where the angles are ignored. The ABCD parameters are derived for line-to-neutral voltages; therefore  $V_S$  and  $V_R$  must also be treated as such.

The real power per pole at the receiving end is:

$$P_R = V_R I_R \quad (2-172)$$

Rewriting (2-172) in terms of  $I_R$  and substituting into (2-167):

$$V_S = R_{DC} \frac{P_R}{V_R} + V_R \quad (2-173)$$

which can also be solved for  $|V_R|$ .

Rewriting (2-173) in terms of (2-126) with  $|V_R|$  as the unknown variable:

$$V_R = \frac{V_S + \sqrt{(V_S^2 - 4R_{DC}P_R)}}{2} \quad (2-174)$$

Losses for DC transmission are dependent on the number of poles (monopolar or bipolar) and the conductor DC resistance. Line losses can be defined as the difference between the sending-end power  $P_S$  and the receiving-end power  $P_R$ :

$$P_{line\ loss} = P_S - P_R \quad (2-175)$$

The sending-end power is unknown and must be calculated:

$$P_S = V_S I_S \quad (2-176)$$

Substituting (2-168), (2-172), (2-176) into (2-175) and simplifying:

$$\begin{aligned} P_{line\ loss} &= V_S I_S - V_R I_R \\ \Rightarrow P_{line\ loss} &= I_{DC} (V_S - V_R) \\ \Rightarrow P_{line\ loss} &= P_R \left( \frac{V_S}{V_R} - 1 \right) \end{aligned} \quad (2-177)$$

## 2.6 TRANSMISSION COSTS

Whenever there are different investment schemes with either different technologies or different scales for an engineering project, a technical and economic quantitative analysis should be carried out and an economic assessment should be given in order to select the optimal scheme [B13]. The economic analysis is done in terms of the *present value of money* (PVM). Present value of money is a method of comparing costs and savings that occur at different times on a consistent and equitable basis for decision-making [B14]. Inflation is not taken into account in the transmission cost equations since it has equal influence on savings and costs.

When comparing transmission costs for an AC or DC interconnection, it is necessary to evaluate not only the investment cost of the line and terminals, but also the cost of losses in order to select the most suitable scheme. Converter losses are another important parameter that must be taken into consideration when calculating DC transmission costs [A7].

In this section the procedure to calculate the transmission costs for both AC and DC technologies will be discussed. Transmissions costs are also referred to in literature as the cost of capital and losses or LCC. The term LCC will be adopted for the purposes of this discussion.

### 2.6.1 Review of the LCC

a) The cost of losses per annum,  $CL(n)$ , can be calculated as:

$$CL(n) = 8760 \cdot ELF \cdot GC \cdot P_{line\ loss} \quad [R] \quad (2-178)$$

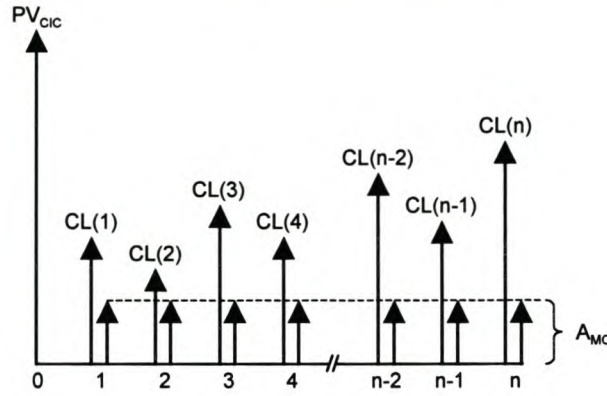
where  $ELF$  = Energy loss factor  
 $GC$  = Generation cost [R/kWh]

Shown in Fig. 2.17 is  $CL(n)$ , which is directly proportional to the square of the annual load factor (ALF) of the line or feeder. ALF is the actual energy supplied over a period divided by the maximum demand over that period multiplied by the time period selected (i.e. actual energy supplied divided by potential energy supplied) [B14]. This proportionality can be quantified by the  $ELF$  [C8] as:



$$ELF = x \cdot ALF + (1 - x) \cdot ALF^2 \quad (2-179)$$

where  $x = 0.2$  (determined by statistical evaluation).



**Fig. 2.17: A graphical representation of the relationship between  $PV_{CIC}$ ,  $A_{MC}$  and  $CL(n)$**

The generation cost, also referred to as the de-escalated energy cost [c/kWh], is fixed for each year and was obtained from the financial department of Eskom.  $P_{line\ loss}$  was determined for AC and DC lines in sections 2.4.4 and 2.5 respectively.

Networks losses ( $P_{Losses}$ ) include the losses in both the line technologies and any additional network support technologies. The cost of compensator losses must be included in the transmission costs of AC networks requiring compensation. Subsequently, converter losses must be taken into account for DC transmission costs. The cost of line and support technology losses can be calculated as follows:

$$\begin{aligned} CL(n) &= 8760 \cdot ELF \cdot GC \cdot (P_{line\ loss} + P_{support\ loss}) \\ \Rightarrow CL(n) &= 8760 \cdot ELF \cdot GC \cdot P_{Losses} \quad [R/yr] \end{aligned} \quad (2-180)$$

where  $P_{support\ loss}$  = kW losses of the support technologies  
 $P_{Losses}$  = Total kW losses

The present value of the annual cost of losses ( $PV_{CL}$ ) can be calculated from the interest rate ( $i$ ) as:

$$PV_{CL}(n) = CL(n) \frac{1}{(1+i)^n} \quad (2-181)$$

where  $n$  is the number of years.

The interest term ( $i$ ) was introduced in (2-181). In the economic assessment of an engineering project the real meaning of interest is different in both concept and value from what is involved in an investment in a bank. It is the result of profit in

the process of the use of capital. In order to differentiate between the two concepts, the term discount rate is substituted for interest rate [B13]. The term discount factor ( $DF$ ) can, therefore, be defined from equation (2-181) as:

$$DF(n) = \frac{1}{(1+i)^n} \quad (2-182)$$

The present value of the total cost of losses ( $PV_{TCL}$ ) can now be calculated over the entire life cycle of the line. This is done by summing all the  $PV_{CL}$  terms calculated in (2-181):

$$\begin{aligned} PV_{TCL} &= \sum_{n=1}^{years} PV_{CL}(n) \\ \Rightarrow PV_{TCL} &= \sum_{n=1}^{years} CL(n) \cdot DF(n) \end{aligned} \quad (2-183)$$

- b) The determination of the annual equivalent maintenance cost ( $A_{MC}$ ) for the line is regularly ignored when doing an economic assessment of AC or DC line technologies. This is because the maintenance cost of one conductor is the same as for another conductor. It would therefore have no significant effect on the reduction of the LCC [C9]. However, when AC and DC line options are economically weighted against each other,  $A_{MC}$  must be considered because the maintenance cost for AC and DC technologies differ. Fig. 2.17 shows the  $A_{MC}$  over the lifetime of the line. The total maintenance cost over the life-cycle of the line can be calculated in terms of the present value of money as:

$$PV_{TMC} = A_{MC} \frac{(1+i)^n - 1}{i(1+i)^n} \quad (2-184)$$

$A_{MC}$  for conventional AC line technologies (no compensation required) will include only the annual maintenance cost of the line:

$$A_{MC} = A_{MC(line)} \quad (2-185)$$

$A_{MC}$  for DC line technologies or AC line technologies with compensation will include the annual maintenance cost of the line and the support technology:

$$A_{MC} = A_{MC(line)} + A_{MC(support)} \quad (2-186)$$



c) The capital investment cost (CIC) is the initial cost of the line and includes the following [C9]:

- Tower structure;
- Foundations;
- Conductor;
- Ground wire;
- Insulation
- Hardware
- Tower erection;
- Stringing;
- Extra work.

The capital investment cost of the two converter stations must be included in the  $PV_{CIC}$  (present value of the CIC) for the DC line technologies. For AC line technologies requiring compensation, the capital investment cost of the support technologies is included. The relationship between  $PV_{CIC}$ ,  $A_{TMC}$  and  $CL(n)$  is shown in Fig. 2.17.

d) The present value of life-cycle cost ( $PV_{LCC}$ ) is calculated from the initial capital investment ( $PV_{CIC}$ ), an annual maintenance component ( $PV_{TMC}$ ) and the cost of losses ( $PV_{TCL}$ ):

$$PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \quad (2-187)$$

### 2.6.2 Determination of the break-even distance

If the transmission distance is great enough, the saving in capital cost and lower losses with DC transmission line may be enough to pay for the two converter stations, one at either end [A8]. This is known as the *break-even distance*. According to Hingorani, the AC vs DC issue in simplistic, generalised terms of break-even distance amounts to [A9]:

- What is the difference in cost between AC and DC lines, taking into account the construction cost and the cost of losses? This difference will favor DC.
- What is the difference in cost between AC substations (including reactive power compensation) and the DC converter stations? This difference will favour AC.



An interesting point is that, as the power requirement is decreased, the saving from AC to DC transmission line per MW-km would increase faster than the increase in the cost of converters, and therefore the break-even distance would generally decrease [A9]. An example is used in the next paragraph to illustrate how the break-even distance is determined.

Consider a 400 kV HVAC line for 2000 MW at a cost of R800 000 per kilometer (or R400 per MW-km). The converter costs of the alternative DC solution are about R1 000 per kW. Allowing R400 per kW for the AC substations, a total of R1200 million must be saved from the line cost to break even. If an HVDC line cost R240 per MW-km, it would take 3750 kilometers of 2000 MW HVDC to break even with AC.

## 2.7 THE APPROPRIATE TECHNOLOGY INDEX (ATI) FOR AC AND DC LINE TECHNOLOGIES

The ATI is used to provide a quantitative measure of the best technical and economic option. The results are not absolute, due to the influence of weighting factors, and are project specific. Key performance indicators (KPIs) are summated using weighting factors to obtain the overall technical and economical performance of a line technology [C9].

The ATI is defined as:

$$ATI = K_1 \cdot w_1 + K_2 \cdot w_2 + K_3 \cdot w_3 + K_4 \cdot w_4 \quad (2-188)$$

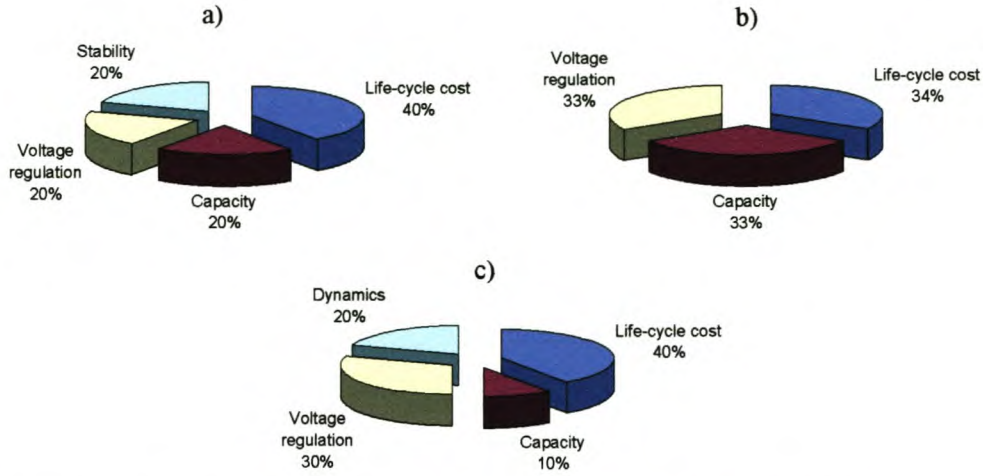
where  $w_1$  to  $w_4$  are weighting factors quantifying the importance of the KPIs,  $K_1$  to  $K_4$ , for a specific project. The KPIs are determined by the technology applied whereas the weighting factors are project specific.

The philosophy of determining the ATI of various line technologies has been successfully applied in the past for HV AC lines at various voltage levels. Recently it has also been applied for MV AC and LV AC lines [C6]. Where compensation is part of the AC line technology, its effect has been included in the various KPIs. The results of these studies are discussed in section 2.7.1.

### 2.7.1 *ATI for AC line technologies*

Fig. 2.18 provides a summary of the KPIs for AC line technologies at the three major voltage levels (HV, MV and LV). At HV levels the KPIs to consider are life-cycle costs, capacity, stability and voltage regulation. MV and LV networks rarely suffer from stability problems and this KPI is therefore ignored for MV and LV levels. Similarly the ATI technique can be applied to LV networks with an added KPI in the form of measuring the dynamic response of the voltage regulator employed to compensate for voltage fluctuations due to sudden load changes.





**Fig. 2.18: KPIs for AC line technologies at various voltage levels: a) HV b) MV and c) LV [C6]**

The first KPI,  $K_1$ , is the total LCC normally taken over a period of 25 years. The calculations of this KPI for conventional AC networks and for compensated AC networks are discussed in detail in section 2.6.1, and only the formula used to represent the total LCC in terms of the PVM will be repeated here:

$$K_1 = PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \quad [R] \quad (2-189)$$

The second KPI,  $K_2$ , is the line cost per probabilistic power rating of the line technology to be compared. This KPI is an indication of the importance of the continuity of supply under heavy-loaded conditions above its previous deterministic thermal current rating [C6]. For conventional AC line technologies:

$$K_2 = \frac{PV_{CI}}{MVA_{emergency}} = \frac{PV_{CIC}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l} \quad [R/km/MVA] \quad (2-190)$$

For AC lines with compensation, the additional capital cost of the support technology must be added to  $P_{CIC}$ . With additional compensation it is also possible to increase the power transfer capacity of the power network. For compensated AC lines:

$$\begin{aligned} K_2 &= \frac{PV_{CI(line)} + PV_{CI(support)}}{MVA_{emergency}} = \frac{PV_{CIC(line)} + PV_{CIC(support)}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l + \Delta S} \\ \Rightarrow K_2 &= \frac{PV_{CI}}{MVA_{emergency}} = \frac{PV_{CIC}}{\sqrt{3} \cdot U \cdot I \cdot N \cdot l + \Delta S} \quad [R/km/MVA] \end{aligned} \quad (2-191)$$

where:

$PV_{CI}$  = present value of the total capital investment (CI) [R/km]  
 $PV_{CI(line)}$  = present value of the CI of the line [R/km]



|                     |   |   |
|---------------------|---|---|
| $PV_{CI(support)}$  | = | present value of the CI of the compensator [R/km] |
| $PV_{CIC}$          | = | present value of the total CIC [R]                |
| $PV_{CIC(line)}$    | = | present value of the CIC of the power line [R]    |
| $PV_{CIC(support)}$ | = | present value of the CIC of the compensator [R]   |
| $U$                 | = | system voltage (line-line) [kV]                   |
| $I$                 | = | ampacity current at 80°C [A]                      |
| $N$                 | = | number of conductors in a bundle                  |
| $l$                 | = | line length [km]                                  |
| $MVA_{emergency}$   | = | emergency loading of the line                     |
| $\Delta S$          | = | apparent power increase due to compensation [MVA] |

The *ampacity*,  $I$ , of a conductor can be defined as the current which will meet the design, security and safety criteria of a particular line on which the conductor is used [A11]. The ampacity can be determined by deterministic or probabilistic means using a number of formulae. A discussion on the deterministic and probabilistic ampacity ratings of conductors is presented in section 3.2.

The third variable,  $K_3$ , is the line cost per surge impedance loading (SIL) of the line technologies to be compared. This variable determines the quality of supply with a stability focus. Some HV lines are not suitable for loading above their SIL limit, because of problems associated with stability due to the fact that the network is mechanically controlled [C6]. This KPI is ignored for MV and LV line technologies, since stability is not a major problem on these networks.

$$K_3 = \frac{PV_{CI}}{MVA_{SIL}} = \frac{PV_{CIC} \cdot Z_e}{U^2 \cdot l} = \frac{PV_{CIC} \cdot \sqrt{\frac{|Z|}{Y}}}{U^2 \cdot l} \quad [\text{R/km/MVA}] \quad (2-192)$$

|                 |   |   |
|-----------------|---|---|
| where $PV_{CI}$ | = | present value of the total capital investment [R/km]      |
| $PV_{CIC}$      | = | present value of the total capital investment cost [R]    |
| $Z_e$           | = | surge impedance of the line [ $\Omega$ ]                  |
| $ Z $           | = | magnitude of the positive sequence impedance [ $\Omega$ ] |
| $Y$             | = | positive sequence [S]                                     |
| $U$             | = | system voltage (line-line) [kV]                           |
| $l$             | = | line length [km]  |

$MVA_{SIL}$  = surge impedance loading of the line

The fourth and last KPI,  $K_4$ , is the percentage voltage drop over the total length of the line. This measures the ability of the particular line technology to meet the set voltage regulation requirements [C6]. Voltage regulation were discussed in great detail in sections 2.4.2 and 2.4.3 and only the equations to calculate the receiving-end voltage ( $V_R$ ) and the voltage regulation ( $V_d$ ) will be repeated here:

$$|V_R| = \sqrt{\frac{F + \sqrt{(F^2 - 4|Z|^2 S_R^2 K)}}{2K}} \quad (2-193)$$

where

$$F = |V_S|^2 - 2RP_R - Q_R(2X - B|Z|^2) \quad (2-194)$$

and

$$K = 1 - BX + \frac{B^2|Z|^2}{4}. \quad (2-195)$$

In (2-194) and (2-195)  $R$  is the total line resistance in per unit,  $X$  the total line inductance in per unit and  $B$  the total line susceptance in per unit.  $P_R$  and  $Q_R$  are respectively the per unit real and reactive power at the receiving-end of the line. Also from (2-193) to (2-195) the following:

$$|Z|^2 = R^2 + X^2 \quad [\Omega] \quad (2-196)$$

From equation (2-132) the voltage regulation of the line can be expressed as:

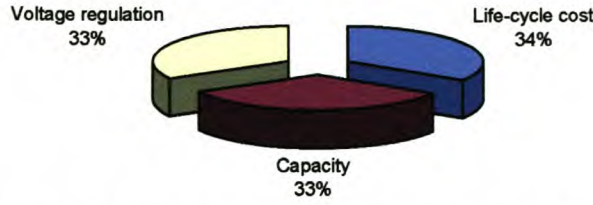
$$V_d = (V_S - V_R) \times 100\% \quad (2-197)$$

where  $V_S$  and  $V_R$  are per unit quantities.

## 2.7.2 ATI for DC line technologies

Fig. 2.19 shows the KPIs for DC line technologies at the three major voltage levels (HV, MV and LV).





**Fig. 2.19: KPIs for DC line technologies at HV, MV and LV levels**

With DC no stability problems occur, because the AC systems are decoupled and the power can be freely and rapidly adjusted by converter control. The KPI ( $K_3$ ) that determines the quality of supply with a stability focus or measures the dynamic response of the converter to compensate for voltage fluctuations at the load end can, therefore, be ignored for DC line technologies. Only the LCC, capacity and voltage regulation KPIs will be considered for DC line technologies.

The calculation of  $K_1$  for DC line technologies are discussed in detail in section 2.6.1 and only the formula used to represent the total LCC in terms of the PVM will be repeated here:

$$K_1 = PV_{LCC} = PV_{CIC} + PV_{TMC} + PV_{TCL} \quad [R] \quad (2-198)$$

The line cost per probabilistic power rating of the DC line technologies to be compared ( $K_2$ ) is derived in similar fashion as its AC counterpart. This KPI is an indication of the importance of the continuity of supply under heavy-loaded conditions. However, overloading is much more restricted in DC transmission, because the silicon-controlled rectifier has a very small thermal capacity and thus the modern converter valve is only designed to handle temporary over-currents. If long-term overload capacity is desired, appropriate overrating of the silicon devices and permitting higher temperatures at the converter plant can achieve this [B2]. For DC line technologies  $K_2$  is defined as:

$$K_2 = \frac{PV_{CI(line)} + PV_{CI(support)}}{MW_{emergency}} = \frac{PV_{CIC(line)} + PV_{CIC(support)}}{2 \cdot U \cdot I \cdot N \cdot I} \quad (2-199)$$

$$\Rightarrow K_2 = \frac{PV_{CI}}{MW_{emergency}} = \frac{PV_{CIC}}{2 \cdot U \cdot I \cdot N \cdot I} \quad [R/km/MW]$$

where:

$PV_{CI}$  = present value of the total capital investment (CI) [R/km]

$PV_{CI(line)}$  = present value of the CI of the line [R/km]



|                     |   |   |
|---------------------|---|---|
| $PV_{CI(support)}$  | = | present value of the CI of the converter [R/km] |
| $PV_{CIC}$          | = | present value of the total CIC [R]              |
| $PV_{CIC(line)}$    | = | present value of the CIC of the power line [R]  |
| $PV_{CIC(support)}$ | = | present value of the CIC of the converter [R]   |
| $U$                 | = | system voltage (pole-neutral) [kV]              |
| $I$                 | = | ampacity current at 80°C [A]                    |
| $N$                 | = | number of conductors in a bundle                |
| $l$                 | = | line length [km]                                |
| $MW_{emergency}$    | = | emergency loading of the line                   |

The voltage regulation KPI ( $K_4$ ) was discussed in great detail in section 2.5 and only the equations to calculate the receiving-end voltage ( $V_R$ ) and the voltage regulation ( $V_d$ ) will be repeated here:

$$V_R = \frac{V_S + \sqrt{(V_S^2 - 4R_{DC}P_R)}}{2} \quad (2-200)$$

where  $R_{DC}$  = total DC line resistance in per unit

$P_R$  = receiving-end real power in per unit

From equation (2-132) the voltage regulation of the line can be expressed as:

$$V_d = (V_S - V_R) \times 100\% \quad (2-201)$$

where  $V_S$  and  $V_R$  are per unit quantities.

### 2.7.3 Implementation of the ATI

In order to equalise the four independent KPIs to one measuring scale, the upper and lower limits per KPI have to be set. The upper limit represents the best-case result (score of 10) and a median value of 3 represents the average of existing solutions. The lower limit represents a score of zero. The rest of the values for each KPI are then linearly extrapolated between the high and low values. This is best illustrated with an example:

**Example:**

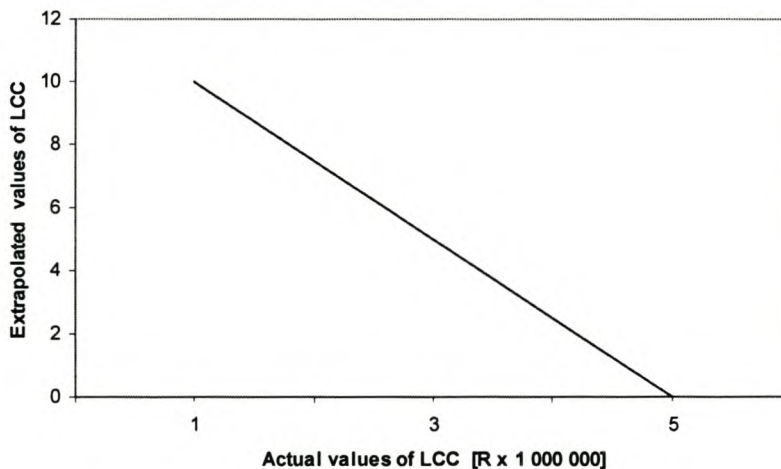
For  $K_1$  the upper limit would be the line technology with the lowest LCC, say R1 000 000, and the lower limit the line technology with the highest LCC, say R5 000 000. The median value can then be calculated as follows:

$$\begin{aligned} \text{Median Value} &= \text{Lower Limit} - \left( \frac{\text{Lower Limit} - \text{Upper Limit}}{10} \right) \cdot 3 \\ \Rightarrow \text{Median Value} &= \text{R5 000 000} - \left( \frac{\text{R5 000 000} - \text{R1 000 000}}{10} \right) \cdot 3 \\ \Rightarrow \text{Median Value} &= \text{R3 800 000} \end{aligned}$$

A value of R3 000 000 is then linearly extrapolated between the low and high values as follows:

$$\begin{aligned} K_{1(\text{extrapolated})} &= 10 - \left( \frac{K_1 - \text{Upper Limit}}{\text{Lower Limit} - \text{Upper Limit}} \right) \cdot 10 \\ \Rightarrow K_{1(\text{extrapolated})} &= 10 - \left( \frac{\text{R3 000 000} - \text{R1 000 000}}{\text{R5 000 000} - \text{R1 000 000}} \right) \cdot 10 \\ \Rightarrow K_{1(\text{extrapolated})} &= 5 \end{aligned}$$

The values above are graphically portrayed in Fig. 2.20.



**Fig. 2.20: Ranking for LCC evaluation**

The weighting factors are used to indicate the dominance of a KPI. The values of the weighing factors are between 0 and 1 (1 = dominant and 0 = no influence at all). If,



for example, the LCC is more important than the voltage drop of a line, then  $w_1$  would be higher than  $w_4$ . The sum of all the weighting factors should be equal to one.

For HV AC line technologies the weighting factors (0.7 0.05 0.05 0.2) mean that the LCC (importance of initial capital expenditure and losses) is 70% dominant, the probabilistic current rating (capacity) and SIL (stability) are 5% dominant and the voltage drop is 20% dominant. The weighting factors are project specific and can be scaled to compare the sensitivity of each factor per chosen line technology. At present the following sensitivities can be evaluated:

- (1,0,0,0) – Normal operation
- (0,1,0,0) – Emergency operation only
- (0,0,1,0) – Surge impedance at HV or dynamic performance at LV levels
- (0,0,0,1) – Voltage drop only

## 2.8 IMPLEMENTATION

TESAT is used to evaluate the voltage regulation, line losses, LCC and ATIs of different AC and DC line technologies. The equations for voltage regulation, line losses, LCC and ATIs discussed in sections 2.4 to 2.7 are implemented in TESAT. These equations were derived using ABCD parameters and trigonometric identities to eliminate the voltage angle ( $\delta$ ), which is usually not known.

The AC interface of TESAT is shown in Fig. 2.21 and the DC interface in Fig. 2.22. The series impedance, shunt conductance, cost and ampacity rating (at 80°C) for three different conductors are specified in the *Compare* frame of the interfaces. Load data, system voltage and line length are also specified, as well as the interest rate, the support technology cost and the ALF. The values of the weighting factors ( $w_1 - w_4$ ) are specified in the *ATI* frame of the interfaces.

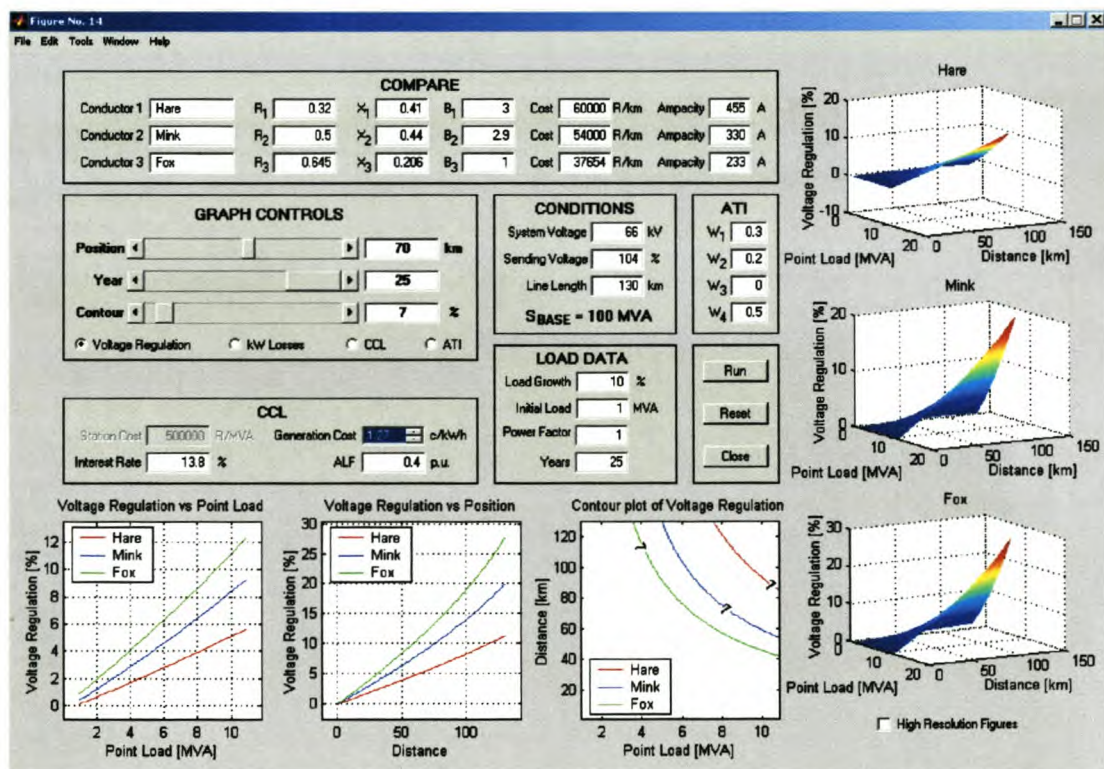


Fig. 2.21: AC interface of TESAT



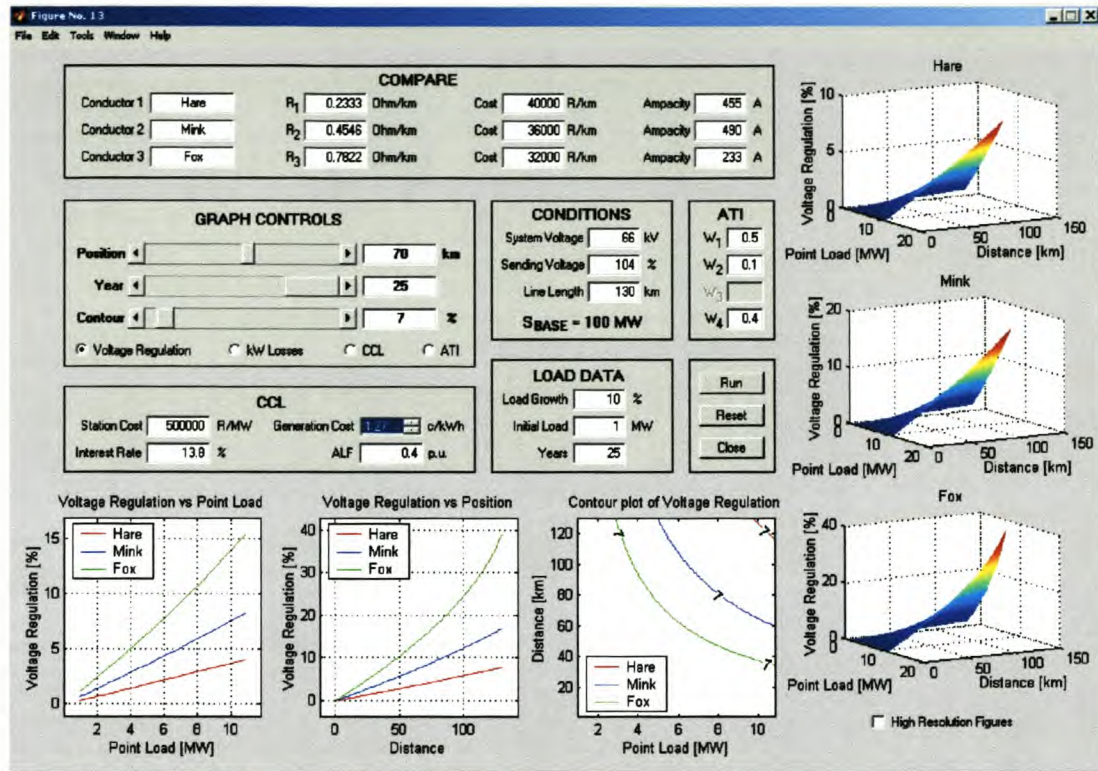


Fig. 2.22: DC interface of TESAT

In Fig. 2.21 and Fig. 2.22 three-dimensional graphs are used to portray the resulting voltage regulation, losses, LCC and ATI for the different AC and DC conductors respectively. The growth in point load is plotted on the X-axis and the line length on the Y-axis. For voltage regulation, losses, LCC and ATIs the magnitude is plotted on the Z-axis. In addition to the three-dimensional graphs three two-dimensional graphs are displayed. These graphs provide cross-sections through the three-dimensional graphs at user-selected points. One cross-section is parallel to the XZ-plane; one is parallel to the YZ-plane and the other parallel to the XY-plane (contour plot). These are labelled as *Voltage Regulation vs. Point Load*, *Voltage Regulation vs. Position* and *Contour plot of Voltage Regulation*, respectively. The user specifies the specific position, the specific year and the specific contour under *Graph Controls*. The parameters of interest (*Voltage regulation*, *kW losses*, *LCC* and *ATI*) are also selectable under *Graph Controls*. Fig. 2.21 and Fig. 2.22 displays the voltage regulation graphs, but it is easy to switch to losses, LCC or ATI graphs.

## Software development

TESAT is implemented, just like OLP, in Matlab. The reasons for choosing Matlab and the use of Matlab GUIs were discussed in section 2.2.4.



The equations to calculate the voltage regulation, line losses, LCC and ATIs for the AC and DC technologies are implemented in *lfac.m* and *lfdc.m* respectively. These programs start by importing all the conductor parameters, system parameters, load parameters and financial parameters. The programs will only function properly if the user enters all the basic parameters before analysis. The same method is followed by *lfac.m* and *lfdc.m* for comparing different line conductors:

1. Calculate the receiving-end voltage and the voltage regulation for the different line technologies from the conductor parameters, system parameters and load parameters specified by the user.
2. With the receiving-end voltage known, the line losses can be calculated for the different lines.
3. With the line losses known, the LCC cost for each line technology can be calculated from the financial parameters.
4. With the LCC costs known, the rest of the ATIs for the different lines can be calculated from the financial parameters.

All the solutions will be real. Line technologies with complex solutions are highlighted in red to indicate that the specific conductor cannot be used. The solutions to the voltage regulation, line losses, LCC and ATIs for each technology are assigned to data matrixes that are used by the GUI to plot the various graphs. The full program listing for *lfac.m* and *lfdc.m* is included in Addendum B4.

TSAT uses basic information to quickly evaluate the voltage regulation, losses, LCC and ATIs of different conductors from an AC and DC perspective in order to select the optimum conductor for the task at hand. TESAT therefore greatly reduces the time and effort involved in the tedious calculations for voltage regulation, losses, LCC and ATIs.



# **CHAPTER 3 LOAD REACH AND TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES**

## **3.1 INTRODUCTION**

## **3.2 THERMAL LOAD REACH – A DETERMINISTIC AND PROBABILISTIC APPROACH**

## **3.3 THERMAL LOAD REACH OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS**

## **3.4 TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS**

## 3.1 INTRODUCTION

Much work has been done recently on the applicability of new line technologies in the stretching of distribution networks. This work included the identification of practical application opportunities for the Electronic Voltage Regulator (EVR) on MV and LV AC distribution networks [A10]. However, the recent advances in the semiconductor technology have also extended the economic power range for DC transmission to just a few MW. With DC transmission the load reach limits of these networks become something of the past. The aim of this chapter is to compare the load reach and transfer capabilities of various AC and DC line technologies as far as bulk energy transfer is concerned.

A discussion on the deterministic and probabilistic current ratings of conductors is presented in section 3.2. In 3.3 the thermal load reach of different AC and DC line technologies are compared at distribution voltage levels: firstly for the Nampower choice of conductors and then for the Eskom choice. It is also shown how deterministic and probabilistic current ratings affect the load reach of these conductors. The transfer capabilities for different AC and DC line technologies, distribution voltages and conductor sizes are presented in section 3.4.



## 3.2 THERMAL LOAD REACH – A DETERMINISTIC AND PROBABILISTIC APPROACH

Willis defines *load reach* as the distance an electrical line can move power before encountering limits in its performance due to engineering criteria [B14]. If the maximum allowable *voltage drop* across a line is set at 7.5% and a line has a voltage drop of 2.5% per kilometer at the recommended loading level, it follows that the line has a load reach of three kilometers or less. If a point load placed at the end of the line requires power to be moved three kilometers or less, then the line is capable of meeting these needs at the recommended loading. However, if the requirement is for power to be moved more than three kilometers, then either the line must be upgraded to one with less voltage drop per kilometer, or the size of the point load must be reduced.

Willis defines thermal load reach *as the distance a particular line type can move power at its thermal loading limit* [B14]. For many years the conductor thermal rating or ampacity has been determined by a deterministic method using conservative ambient conditions. It is a quick and simple method and has been used by Eskom and other utilities around the world. The deterministic approach assumes certain bad cooling conditions (low wind speed, high ambient temperature, etc.) and calculates the current that would result in the line design temperature being reached. The line templating or design temperature is that temperature at which the height of the conductor above the ground is the minimum permissible [A11]. The drawback of this method is that it does not address the safety or the relationship between safety and the power transfer capability.

However, in today's economic environment it is necessary to use assets more effectively and, on power lines, costs can be deferred or saved by finding ways to operate lines closer to their thermal limits. A new approach to that of weighing up the probability of risk of flashover due to environmental conditions and ground clearances has resulted in the so-called probabilistic ampacity ratings of conductors. The probabilistic approach provides the means to calculate the thermal rating at different templating temperatures. It also quantifies the probability of an unsafe condition arising associated with the rating and keeps this constant for conductors of a



similar type. The probabilities applied are based on present probabilities so that if the line is utilised at a higher temperature, the probability of an unsafe condition arising is no more than at present.

The deterministic approach will be used for the Nampower choice of conductors and the probabilistic approach for Eskom conductor choices. The reason for this is firstly to illustrate the use of both the deterministic and probabilistic method to establish the table of conductor ampacities. Secondly, the probabilistic values at different template temperatures for some of the Nampower conductor choices are not readily available.

The ampacity values for the Nampower conductor selection at distribution voltage levels are shown in Table 3-1. The conductors in Table 3-1 are thermally rated for a 75°C electrical rating, but the line template temperature is at 50°C.

**Table 3-1: Conductor ampacity table using deterministic loading techniques [C12]**

| Conductor<br>Type | Templating<br>Temperature [°C] | Normal Loading (75°C) |                  | Emergency Loading (90°C) |                  |
|-------------------|--------------------------------|-----------------------|------------------|--------------------------|------------------|
|                   |                                | $I = I_{AC}$ [A]      | $I = I_{DC}$ [A] | $I = I_{AC}$ [A]         | $I = I_{DC}$ [A] |
| Gopher            | 50                             | 122                   | 128              | 147                      | 154              |
| Fox               | 50                             | 150                   | 157              | 181                      | 190              |
| Hare              | 50                             | 285                   | 297              | 349                      | 363              |
| Rabbit            | 50                             | 186                   | 196              | 225                      | 238              |
| Oak               | 50                             | 288                   | 288              | 370                      | 370              |

Table 3-2 shows the probabilistic ampacity values for the conductors used by Eskom at 11, 22 and 33 kV. For the purpose of generating the table the probability of an unsafe condition occurring was calculated for the Eskom design philosophy prior to 1987 [C11]. With this philosophy the conductor was thermally rated for a 75°C electrical rating, but the line template temperature was at 50°C. In the equations the probability of an unsafe condition was then kept constant and the ratings at different template temperatures were calculated.



**Table 3-2: Conductor ampacity table using probabilistic loading techniques [C13], [C12]**

| Conductor Type | Templating Temperature [°C] | Normal Loading   |                  | Emergency Loading |                  |
|----------------|-----------------------------|------------------|------------------|-------------------|------------------|
|                |                             | $I = I_{AC}$ [A] | $I = I_{DC}$ [A] | $I = I_{AC}$ [A]  | $I = I_{DC}$ [A] |
| Magpie         | 50                          | 33               | 35               | 40                | 42               |
| Magpie         | 60                          | 47               | 49               | 52                | 55               |
| Magpie         | 70                          | 58               | 61               | 62                | 65               |
| Magpie         | 80                          | 67               | 70               | 70                | 73               |
| Magpie         | 90                          | 74               | 78               | 77                | 81               |
| Magpie         | 100                         | 81               | 85               | 84                | 88               |
| Squirrel       | 50                          | 106              | 111              | 135               | 142              |
| Squirrel       | 60                          | 130              | 136              | 160               | 168              |
| Squirrel       | 70                          | 149              | 156              | 181               | 190              |
| Squirrel       | 80                          | 165              | 173              | 198               | 208              |
| Squirrel       | 90                          | 178              | 187              | 213               | 223              |
| Squirrel       | 100                         | 190              | 199              | 227               | 238              |
| Fox            | 50                          | 148              | 155              | 192               | 201              |
| Fox            | 60                          | 184              | 193              | 228               | 239              |
| Fox            | 70                          | 210              | 220              | 258               | 271              |
| Fox            | 80                          | 233              | 244              | 283               | 297              |
| Fox            | 90                          | 253              | 265              | 305               | 320              |
| Fox            | 100                         | 270              | 283              | 324               | 340              |
| Mink           | 50                          | 209              | 220              | 272               | 286              |
| Mink           | 60                          | 258              | 271              | 324               | 341              |
| Mink           | 70                          | 297              | 313              | 367               | 386              |
| Mink           | 80                          | 330              | 347              | 402               | 423              |
| Mink           | 90                          | 357              | 376              | 434               | 457              |
| Mink           | 100                         | 382              | 402              | 461               | 485              |
| Hare           | 50                          | 292              | 304              | 380               | 396              |
| Hare           | 60                          | 357              | 372              | 454               | 473              |
| Hare           | 70                          | 408              | 425              | 515               | 536              |
| Hare           | 80                          | 455              | 474              | 565               | 588              |
| Hare           | 90                          | 496              | 516              | 609               | 634              |
| Hare           | 100                         | 529              | 551              | 685               | 713              |

It is important to note that if the values in Table 3-2 are to be used, it is of vital importance that the template temperature and the conductor thermal rating are the same, e.g. for a 50°C template temperature the normal and emergency values at 50°C electrical rating must be used. If the template temperature and conductor thermal

rating are different the probability of an unsafe condition will not be the same as that for the calculated values in Table 3-2.

The conductor types and sizes specified in Table 3-1 and Table 3-2 have already been standardised by both Nampower and Eskom at distribution level and are adopted without change to this work [C14],[C15]. The conductor properties were sourced from the data sheets of overhead aluminium conductors and are included in Addendum A.1.



### 3.3 THERMAL LOAD REACH OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS

The receiving-end voltage for three-phase AC lines was calculated in Chapter 2 as a function of the sending-end voltage, the receiving-end power and the impedance. From equation (2-127) the receiving-end voltage can be written as:

$$\begin{aligned} |V_S|^2 - 2r_{ac}IP_R - Q_R(2xl - bl(r_{ac}^2l^2 + x^2l^2)) = \frac{(r_{ac}^2l^2 + x^2l^2)S_R^2}{|V_R|^2} \\ + \left[ 1 - blxl + \frac{b^2l^2(r_{ac}^2l^2 + x^2l^2)}{4} \right] |V_R|^2 \end{aligned} \quad (3-1)$$

where  $r_{ac}$  and  $x$  is the series impedance per kilometer and  $b$  the shunt admittance per kilometer.

The thermal load reach can be calculated by rewriting equation (3-1) in terms of the line length. The solution to the line length is a fourth-order solution. Explicit solutions exist for fourth-order equations, but the formulae are very long and tedious to work with. Maple, a symbolic mathematical software package from Waterloo, was used to solve (3-1) in terms of the line length  $l$ :

$$\sqrt{c_1l^4 + c_2l^3 + c_3l^2 + c_4l + c_5} = 0 \quad (3-2)$$

where

$$c_1 = V_R^4 b^2 (r^2 + x^2) \quad (3-3)$$

$$c_2 = -4Q_R V_R^2 b (r^2 - b^2) \quad (3-4)$$

$$c_3 = 4(S_R^2 r^2 + S_R^2 x^2 - V_R^4 bx) \quad (3-5)$$

$$c_4 = 8V_R^2 (rP_R + xQ_R) \quad (3-6)$$

$$c_5 = 4V_R^2 (V_R^2 - V_S^2) \quad (3-7)$$

Matlab was used to calculate the values for the constants  $c_1 - c_5$  and the line length  $l$ . Addendum B5 contains the listing of *constants.m*, the file written to solve (3-2). Four possible solutions exist for the line length, and the correct solution must be chosen from the four possibilities. Two of the solutions to (3-2) are complex numbers and

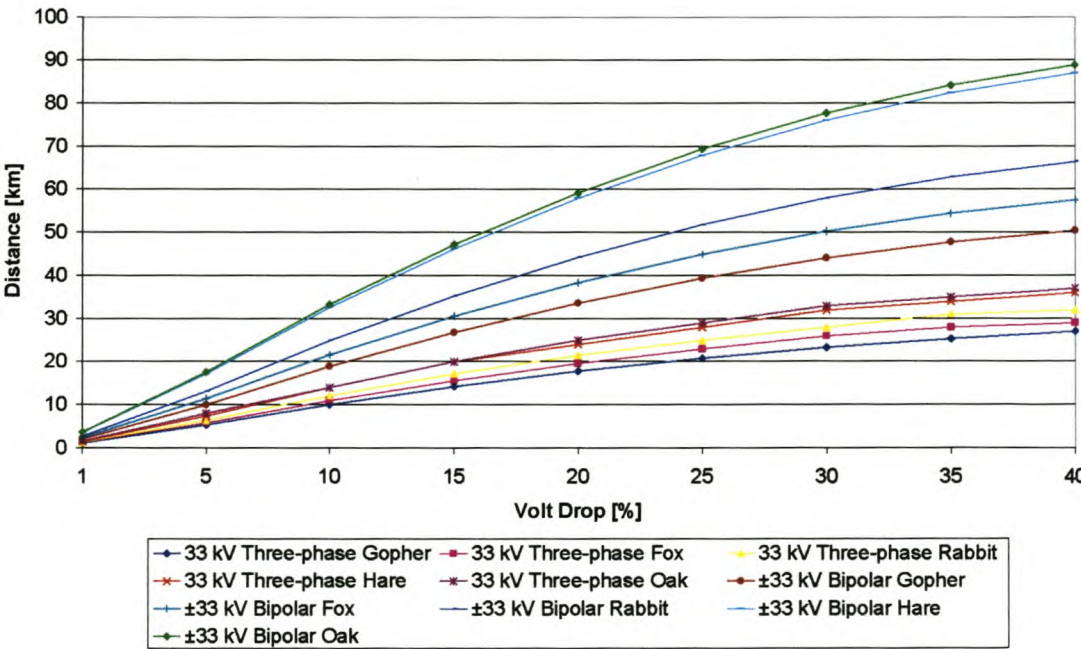
the other two are real numbers, of which one is a negative number. Both the complex solutions and the negative solution are ignored, as line length cannot be a complex or negative number.

Similarly the thermal load reach can be calculated for bipolar DC lines by rewriting equation (2-173) in terms of the line length:

$$l = \frac{V_R(V_S - V_R)}{r_{dc}P_R} \tag{3-8}$$

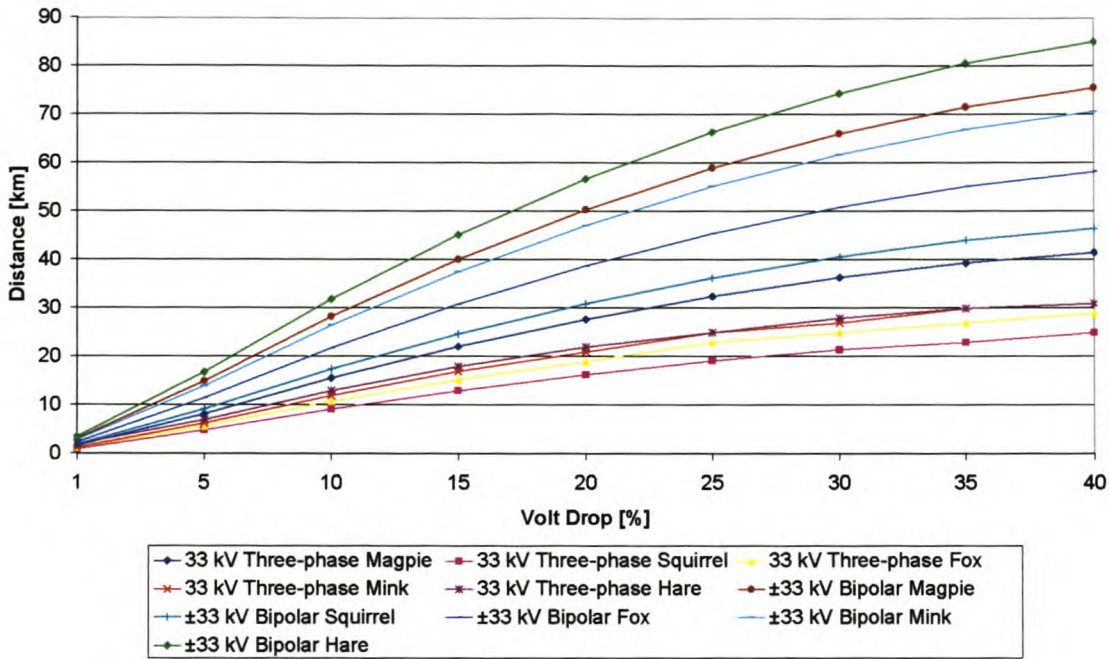
where  $r_{dc}$  is the dc resistance per kilometer.

Fig. 3.1 shows the deterministic thermal load reach of 33 kV three-phase AC and bipolar DC networks for various Nampower conductor types and sizes. Fig. 3.2 shows the probabilistic load reach of these networks for different Eskom conductors at a template temperature of 50°C. A power factor of 0.95 was assumed for the AC networks.



**Fig. 3.1: Deterministic thermal load reach of different Nampower conductors from an AC and DC perspective**

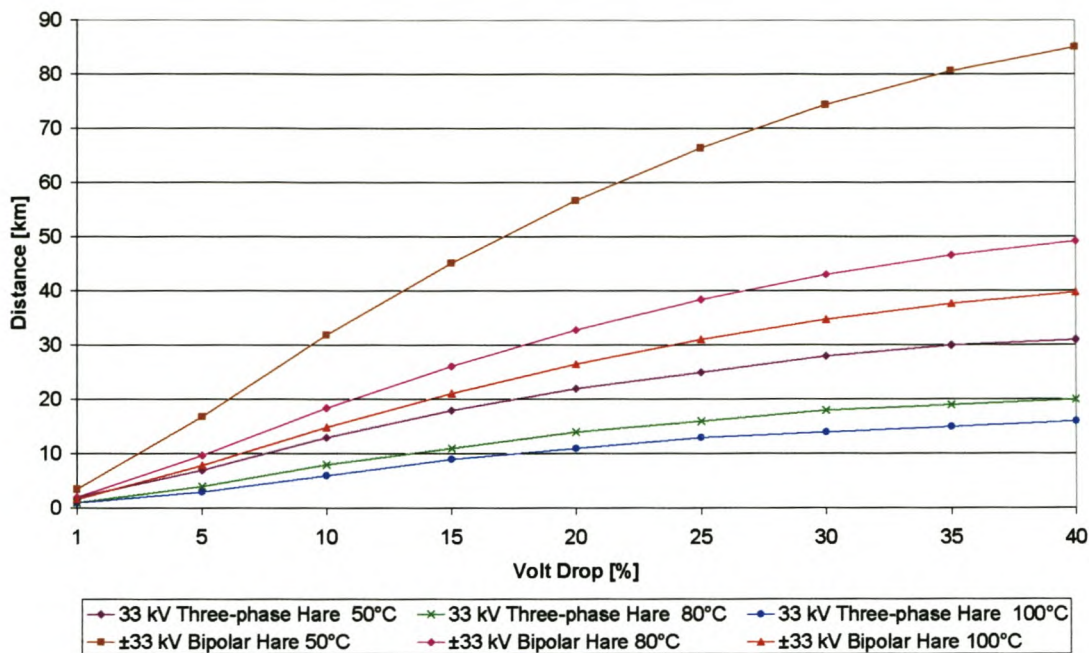




**Fig. 3.2: Probabilistic thermal load reach of different Eskom conductors from an AC and DC perspective**

Fig. 3.1 and Fig. 3.2 show that a reduction in voltage-drop restrictions implies an increased conductor length by more than three to four times. Cost can be significantly reduced on medium and low voltage AC networks by increasing the voltage regulation margin by more than 30% [A10]. Cost comparisons on different options to compensate for these low voltages were done. The study not only investigated the applicability of new line technologies in the stretching of AC distribution networks, but also innovative techniques for network support to establish longer load-reach capabilities. The cost savings were evaluated against the cost of boosting the reduced voltage levels to the required margins [A12].

The voltage drop limitations of AC networks become something of the past when applying DC line technologies. The voltage source converter at the load-end is able to compensate for a voltage drop of more than 30% with no commutation problems (transistor-based technology). Also the lower losses with DC transmission line imply longer load reach distances, as shown in Fig. 3.1 and Fig. 3.2. However, because it requires two converter stations having a total power rating of twice the maximum required power transmission, DC technology is not that inexpensive. The cost savings must, therefore, be evaluated against the cost of the two converter stations.



**Fig. 3.3: Probabilistic load reach of 33 kV three-phase and bipole Hare at different template temperatures**

Shown in Fig. 3.3 is the combined impact of increased current ratings (at template temperatures of 50, 80 and 100°C) and extended voltage-regulation margins in determining the thermal load reach of 33 kV three-phase and bipolar Hare. By increasing the template temperature the ratings of the different lines increase, but this in turn implies shorter load reach distances.



### 3.4 TRANSFER CAPABILITIES OF AC AND DC LINE TECHNOLOGIES AT DISTRIBUTION VOLTAGE LEVELS

An analysis has been made to illustrate the effectiveness of different MV line technologies from a load capacity [MVA] and transfer capability [MVA.km] point of view [C16].

The transfer capability of a line technology is basically the amount of power that can be transferred over one kilometer of the line, at a specific voltage regulation limit and power factor. Rewriting equation (2-127) in terms of the receiving-end power  $S_R$  and substituting line length ( $l$ ) with one kilometer, the transfer capabilities of three-phase AC line technologies at distribution level can be determined as:

$$0 = c_6 S_R^2 + c_7 S_R + c_8 \quad (3-9)$$

where

$$c_6 = (r^2 + x^2) = |z|^2 \quad (3-10)$$

$$c_7 = (2r \cos \theta + (2x - b|z|^2) \sin \theta) |V_R|^2 \quad (3-11)$$

$$c_8 = \left( 1 - bx + \frac{b^2(r^2 + x^2)}{4} \right) |V_R|^4 - |V_S|^2 |V_R|^2 \quad (3-12)$$

Equation (3-9) can be solved with the general expression for a quadratic equation:

$$MVA \cdot km = \frac{-c_7 \pm \sqrt{c_7^2 - 4c_6 c_8}}{2c_6} \quad (3-13)$$

which will yield two possible solutions. The negative solution is ignored, as the transfer capability of a line cannot be a negative number. Again Matlab was used to calculate the values for the constants  $c_6 - c_8$  and the transfer capability  $MVA \cdot km$ . Addendum B6 contains the listing of *transfer.m*, the file written to solve (3-13).

The transfer capability of a bipolar DC line can easily be calculated from equation (2-173), where the line length is taken as one kilometer:



$$MW \cdot km = \frac{V_R(V_S - V_R)}{r_{dc}} \tag{3-14}$$

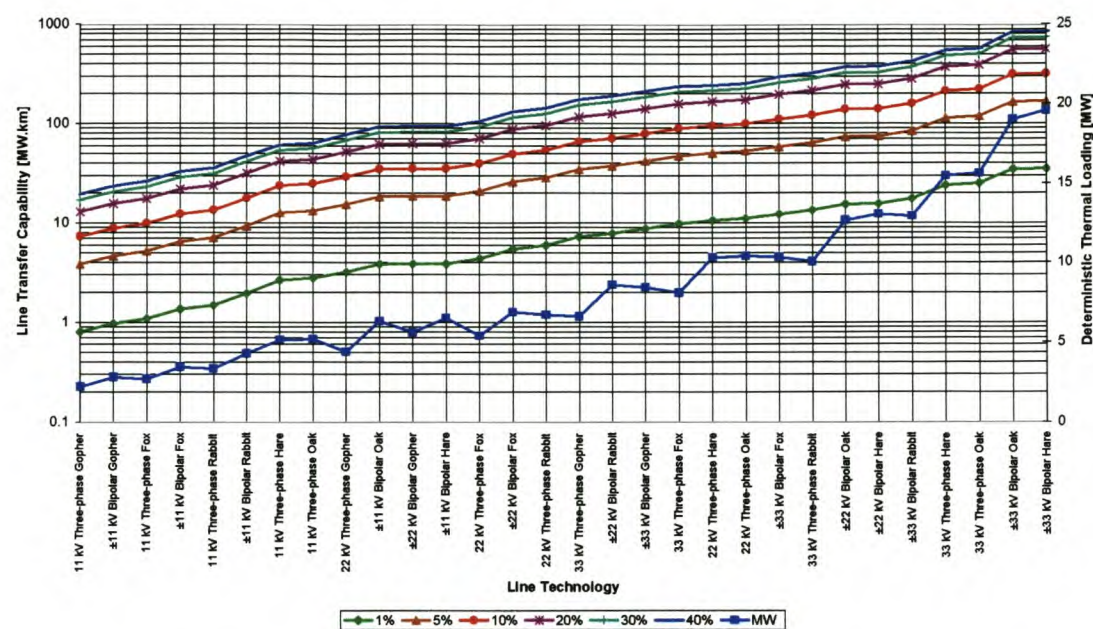


Fig. 3.4: Line transfer capabilities for different Nampower AC and DC line technologies, distribution voltages and conductor sizes

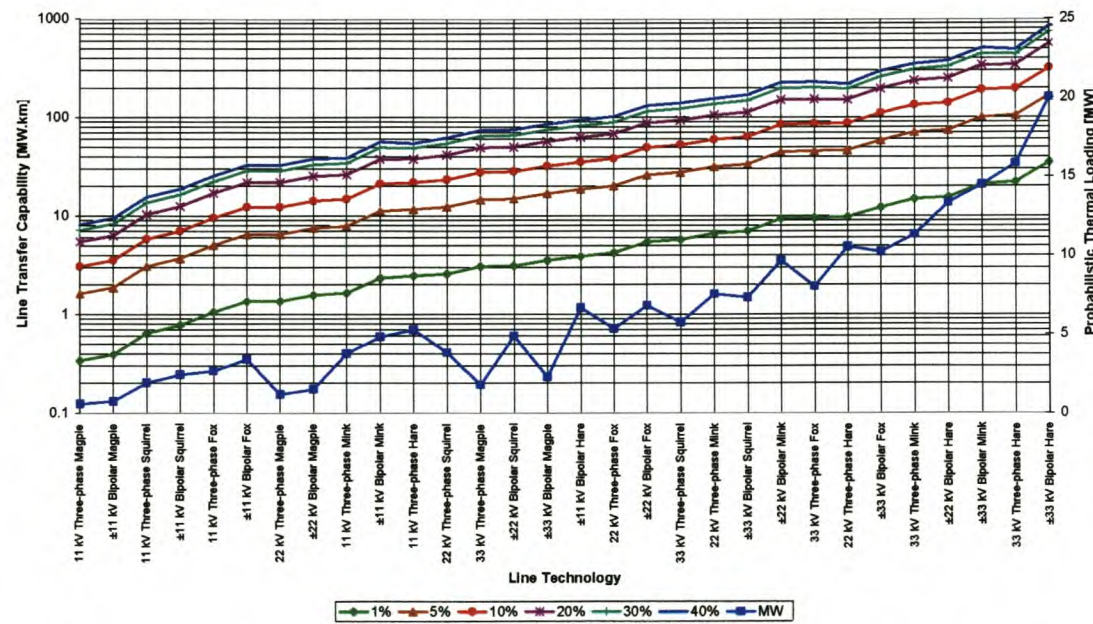


Fig. 3.5: Line transfer capabilities for different Eskom AC and DC line technologies, distribution voltages and conductor sizes

Fig. 3.4 and Fig. 3.5 summarize the line transfer capabilities of different three-phase and bipolar networks at distribution voltage levels for Nampower and Eskom conductor sizes, respectively. A power factor of 0.95 was used to calculate the real power transfer capabilities of the AC line technologies.



These graphs show that higher voltage lines can transfer more power, since power is proportional to the square of the voltage. They normally cost more due to expensive insulation, but this is offset by the much increased transfer capability [C6]. Also shown in Fig. 3.4 and Fig. 3.5 is the much higher transfer capability of bipolar DC lines for the same conductor size. Even if the voltage drop limitations of AC distribution networks is increased beyond the acceptable norm of 7.5% and 10% up to 40% [B15], DC lines will still have superior transfer capabilities over AC lines, due to lower line losses.

# **CHAPTER 4 THE KATIMA MULILO CASE STUDY**

## **4.1 INTRODUCTION**

## **4.2 THE KATIMA MULILO CASE STUDY**

### **4.2.1 AC line technology optimisation**

### **4.2.2 Obtaining network parameters**

### **4.2.3 Uncompensated network**

## **4.3 SHUNT COMPENSATION**

## **4.4 SERIES COMPENSATION**

## **4.5 SERIES-SHUNT COMPENSATION**

## **4.6 IN-LINE COMPENSATION – THE DC ALTERNATIVE**

### **4.6.1 DC line technology optimisation**

### **4.6.2 Simplified network**

## **4.7 SUMMARY OF THE KATIMA MULILO CASE STUDY**



## 4.1 INTRODUCTION

Chapter 4 represents the practical application of the theory developed in Chapter 2 and Chapter 3. A case study is presented which involves the construction of a new line to transmit a small amount of power over a long distance. Two transmission technologies are evaluated, namely 132 kV three-phase AC and  $\pm 50$  kV bipolar DC. It is shown that the AC network will require compensation due to overvoltages at a distance from the source and the effects of shunt, series and series-shunt compensation are considered for these networks. The DC network is evaluated as an in-line compensated AC network.

A description of the Katima Mulilo case study is presented in section 4.2. Different ACSR conductors are evaluated by TESAT in 4.2.1 to determine the optimum conductor for a 500 km, 132 kV three-phase AC line. The methods used to determine the network parameters for the generic network of PSAT are also presented in 4.2.2. A voltage regulation problem for the AC network is identified in 4.2.3 and in sections 4.3 - 4.5, shunt, series and series-shunt devices are analysed as voltage regulators. In section 4.6.1 different ACSR conductors are evaluated by TESAT to determine the optimum conductor for a 500 km,  $\pm 50$  kV bipolar DC line. PSAT is then used to analyse the voltage regulation performance of the DC transmission network as an in-line technology in 4.6.2.

The chapter is concluded with a summary of the different options available for constructing the new network in order to determine the optimum solution for this case study.

## 4.2 THE KATIMA MULILO CASE STUDY

The Caprivi region, shown in Fig. 4.1, is a thin projection of north-eastern Namibia surrounded by the countries of Angola, Zambia and Botswana. Its environmental character derives from the three perennial rivers that cross and border the area and the flat landscape that separates them.

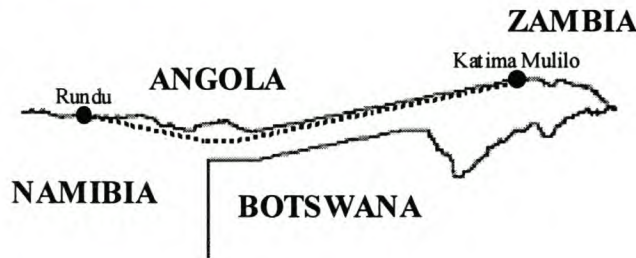


Fig. 4.1: The Caprivi region

The current total electrical load of the Caprivi Region amounts to approximately 0.52 MVA at a power factor of 0.96. This load is supplied from Katima Mulilo distribution station. The Caprivi network is isolated from the rest of the Namibian electricity network (about 500 km from the closest point of supply at Rundu). The Katima Mulilo network is currently connected to the Zambian national electricity supply authority (ZESCO). The Zambian network is heavily loaded and operates at the limits of its capacity. This results in frequent power outages. The total load of the Caprivi region is anticipated to reach 8.5 MW after 20 years, which corresponds to a load growth of 15.22% per annum [C17].

### 4.2.1 AC line technology optimisation

The AC interface of TESAT is used to evaluate three different Nampower conductor choices in order to select the optimum conductor for a 500 km, 132 kV three-phase AC line (shown in Fig. 4.1) between Rundu substation and Katima Mulilo distribution station. The following three line technologies were evaluated:

- Wolf conductor with a TRANS 980 tower
- Pelican conductor with a Concrete tower
- Twin Hare conductor with a Kamerad tower.

The line structures used by Nampower for the above-mentioned conductors are shown in Fig. 4.2. The data for the different conductors were sourced from conductor property tables and are included in Addendum A1.



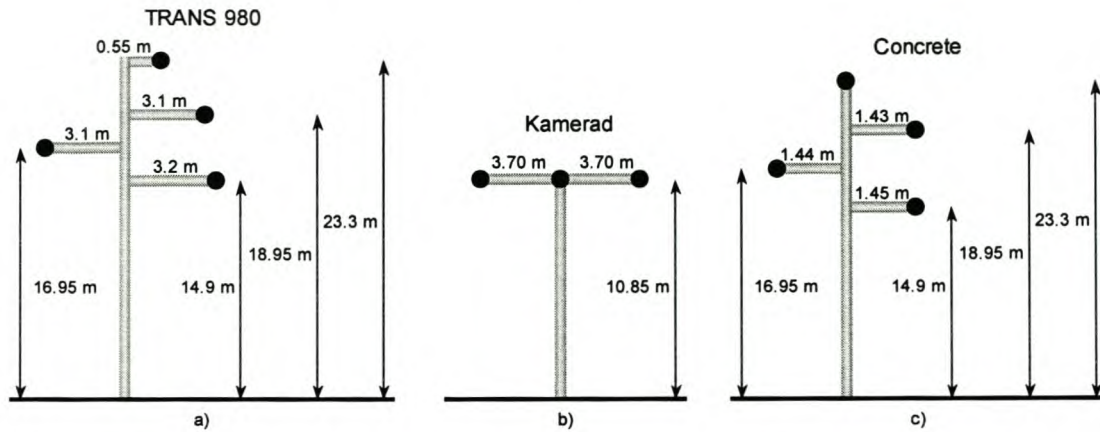


Fig. 4.2: 132 kV Nampower line structures [C18]

The series impedance and shunt admittance of the different conductors and line structures were calculated, using the OLP interface of PSAT. The RXB values for the different line technologies are given in Table 4-1, as well as the deterministic ampacity values and the cost of these lines. The AC resistance values at 40°C were calculated from the DC resistance values at 20°C.

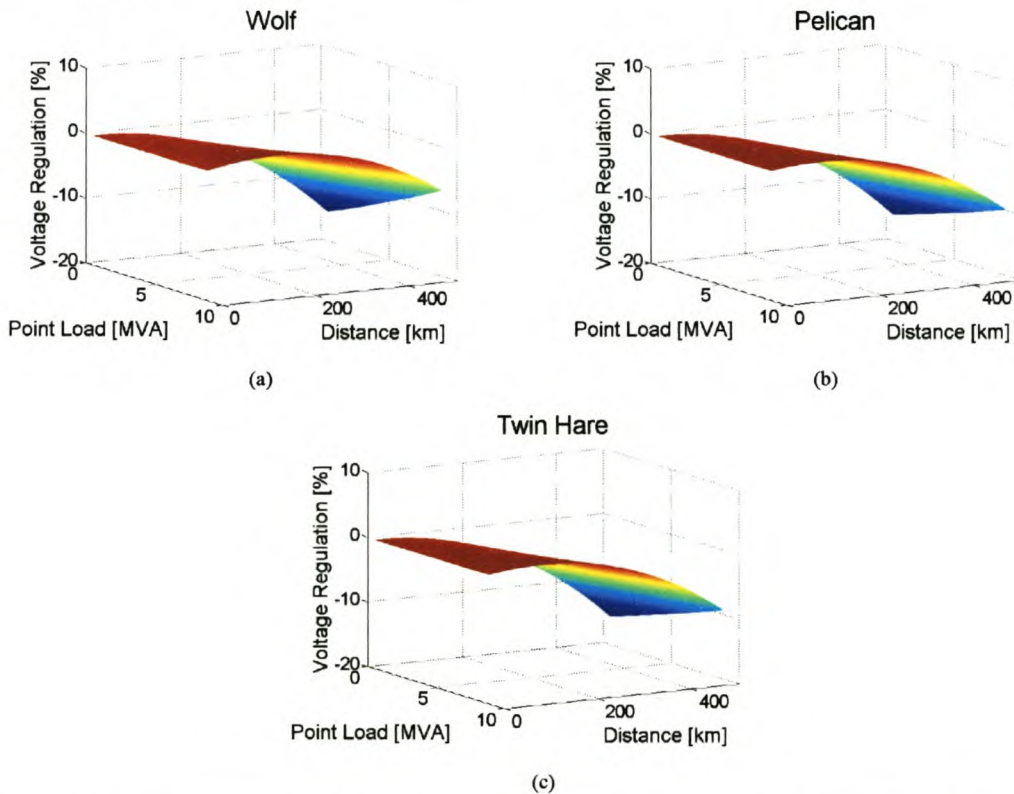
Table 4-1: 132 kV Nampower line technology data

| Conductor | Tower     | $r_{ac} (40^{\circ}\text{C})$<br>[ $\Omega/\text{km}$ ] | $X$<br>[ $\Omega/\text{km}$ ] | $B$<br>[ $\mu\text{S}/\text{km}$ ] | Ampacity<br>[A] | Line Cost<br>[R/km] |
|-----------|-----------|---|-------------------------------|------------------------------------|-----------------|---------------------|
| Wolf      | TRANS 980 | 0.2091  | 0.4157                        | 2.72                               | 372             | 127 000             |
| Pelican   | Concrete  | 0.1291  | 0.3665                        | 3.13                               | 486             | 151 000             |
| Twin Hare | Kamerad   | 0.1601  | 0.3041                        | 3.73                               | 571             | 179 000             |

The data in Table 4-1 are entered in the AC interface of TESAT. A base value of 100 MVA is used for  $S_{base}$  and 132 kV for  $V_{base}$ . Eskom supplied the generation cost for the next 20 years, where the year 1999 was taken as the base year. The current interest rate was taken as 13.8% and the ALF as 0.4. Nampower supplied the line costs, which are treated as confidential and are used only for reference purposes. The costs of the line technologies per kilometre were calculated using a cost estimation tool provided by Eskom. These costs included material costs, construction costs, survey costs, field services, engineering fees and general overhead costs.

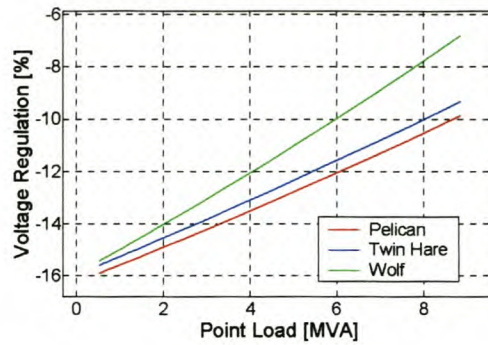
## Voltage regulation analysis

The voltage plots indicating the voltage regulation of the different line technologies are shown in Fig. 4.3. The growth in point load is plotted on the X-axis and the physical position along the line on the Y-axis. Therefore, the Z-axis represents the voltage magnitude. These graphs can be used for additional insight into the voltage regulation of different line technologies for different point load sizes and distances. From Fig. 4.3 it is clear that the three-phase line technologies will suffer from voltage regulation problems.



**Fig. 4.3: Voltage regulation of the AC line technologies as a function of distance and load growth**  
A cut parallel to the XZ-plane at a distance of 500 km (Fig. 4.4) reveals the true extent of the voltage regulation problem. Due to the long distance and the small point load, there is a rise in the receiving-end voltage, due to the flow of capacitive (line-charging) current through line inductance (Ferranti effect). The worst-case scenario is a sudden opening of the line at the receiving end, which will cause the sending-end voltage to rise due to the capacitive current of the line flowing through the source impedance. This worst-case over-voltage is approximately 16% (Fig. 4.4).





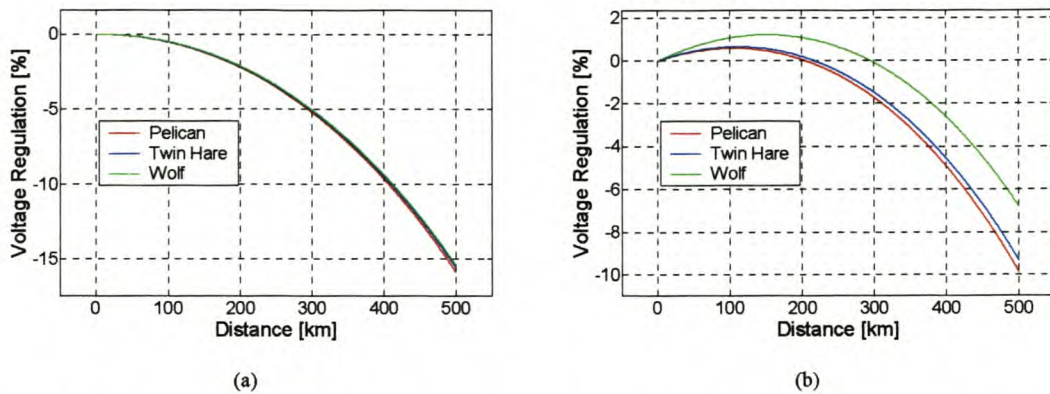
**Fig. 4.4: Voltage regulation of the AC line technologies as a function of load growth**

Table 4-2 provides a summary of the voltage regulation performance for the different AC line configurations in Fig. 4.4. The values in Table 4-2 are well above the NRS 048 standard of 1.05 p.u. for  $\geq 500$  V networks [B16]. Appropriate forms of line compensation must therefore be provided to keep the overvoltages on the lines to acceptable limits.

**Table 4-2: Summary of the voltage regulation performance**

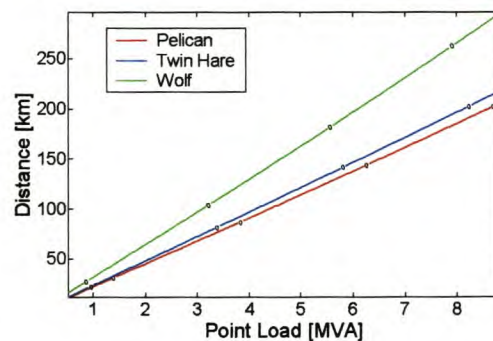
| 132 kV<br>Three-phase AC | % Voltage Regulation |         |
|--------------------------|----------------------|---------|
|                          | Minimum              | Maximum |
| Pelican                  | -9.8                 | -15.9   |
| Twin Hare                | -9.3                 | -15.6   |
| Wolf                     | -6.8                 | -15.4   |

A cut parallel to the YZ-plane gives the voltage regulation at different positions along the line for a specific load. The voltage regulation for an initial load of 0.52 MVA and a final load of 8.85 MVA is shown in Fig. 4.5. Fig. 4.5 shows that, as the load grows from its initial value to its final value, there will be an improvement in the voltage regulation along the line. Depending on the load current, the lines will either absorb or supply reactive power. At loads below the SIL the lines will produce net reactive power. Increasing the load increases the line current that flows from the source to the load, decreasing the net effect of the line charging current and improving the voltage regulation along the line.



**Fig. 4.5: Voltage regulation of the AC line technologies as a function of position for an (a) initial load of 0.52 MVA and (b) a final load of 8.85 MVA**

The voltage regulation of the different line technologies can also be explained with the aid of a contour plot, which is a cut parallel to the XY-plane. Fig. 4.6 shows the voltage regulation contours of the different line technologies for a value of zero regulation. The growth in point load is plotted on the X-axis and the position along the line on the Y-axis. The value for voltage regulation will be negative (rise in receiving-end voltage) for all the combinations of point load and position above the voltage regulation contours of the different line technologies. For all the values below the contours the voltage regulation will be positive.



**Fig. 4.6: Contour plot for voltage regulation of the AC line technologies**

As for voltage regulation, TESAT provides similar three- and two-dimensional graphs for power losses, transmission costs and ATIs. The graphs were analysed in detail for voltage regulation, and from this point forward only the graphs portraying voltage regulation, losses, transmission costs and ATIs against growth in point load will be studied.

## Transmission losses

The power losses in kW for the different three-phase line technologies are shown in Fig. 4.7 and do not include compensation losses. The lines will experience maximum



power losses for initial load conditions. The maximum losses for the different lines are more than 100% of the transmitted power and are due to large line charging currents flowing through the line inductance for small loads over long distances. As the load increases, the line charging-currents and the power losses decreases to a minimum value that differs for the three cases shown in Fig. 4.7.

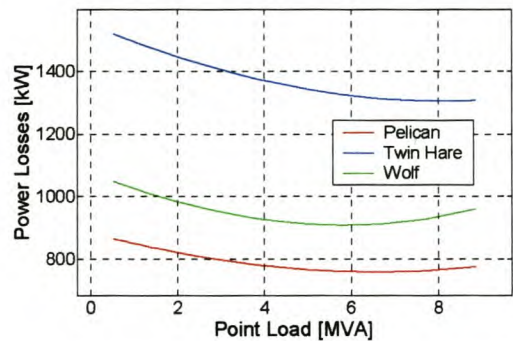


Fig. 4.7: Power losses of the AC line technologies as a function of load growth

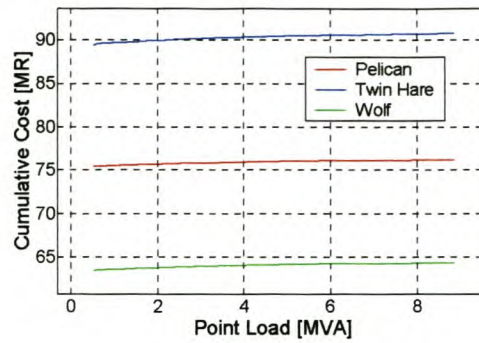
Table 4-3 provides a summary of the power losses for the different AC line configurations in Fig. 4.7. The maximum losses are between 150% and 260% of the transmitted power, whereas the minimum losses are between 11% and 17% of the transmitted power.

Table 4-3: Summary of power losses

| 132 kV<br>Three-phase AC | Minimum           |                  | Maximum           |                  |
|--------------------------|-------------------|------------------|-------------------|------------------|
|                          | $P_{phase\ loss}$ | $P_{line\ loss}$ | $P_{phase\ loss}$ | $P_{line\ loss}$ |
|                          | [kW]              | [kW]             | [kW]              | [kW]             |
| Pelican                  | 253               | 760              | 289               | 867              |
| Twin Hare                | 436               | 1 307            | 507               | 1 522            |
| Wolf                     | 303               | 910              | 350               | 1 050            |

Transmission costs

Fig. 4.8 shows the transmission costs or LCC of the different three-phase line technologies in terms of the present value of money. The cost of line compensation required for the AC line technologies is not included in the LCC of Fig. 4.8. The three-phase Wolf proves to be the most economic solution for the lifetime of the AC network (20 years), while Twin Hare will be the most expensive solution.



**Fig. 4.8: LCC for the AC line technologies**

Table 4-4 provides a summary of the minimum and maximum present values of the LCC over the lifetime of the uncompensated AC network. The present value of the cost of losses is also shown in Table 4-4, where Pelican will have the lowest cost of losses and Twin Hare the highest cost of losses.

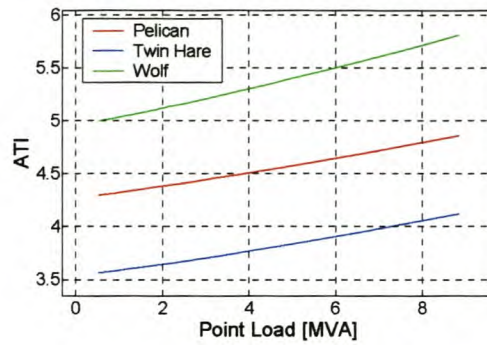
**Table 4-4: Summary of LCC and cost of losses**

| 132 kV<br>Three-phase AC | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|--------------------------|-----------------|---------|-----------------|
|                          | Minimum         | Maximum |                 |
| Pelican                  | 75.50           | 76.27   | 0.77            |
| Twin Hare                | 89.50           | 90.85   | 1.35            |
| Wolf                     | 63.50           | 64.43   | 0.93            |

### Appropriate technology index (ATI)

The ATI technique is used to identify the optimum conductor for the three-phase line. As was discussed previously in section 2.7.1., the KPIs to consider for HV networks are LCC, capacity, stability and voltage regulation. The capacity KPI is ignored, because the load current requirement is much smaller than the current rating of the conductors used to supply the load. Owing to small point load, the power angle across the line is small (about  $9^\circ$ ), and as a result the stability KPI is also ignored. All the line technologies will require voltage compensation during the lifetime of the network and, therefore, the weighting factors for the LCC and voltage regulation are chosen to be 50% each.





**Fig. 4.9: ATI of the three-phase line technologies**

The ATIs for the three-phase line technologies are shown in Fig. 4.9 and the results are summarised in Table 4-5. Fig. 4.9 clearly indicates Wolf conductor to be the optimum choice for the lifetime of the line. On the contrary, Twin Hare conductor would be the least optimum choice to string the line with.

**Table 4-5: ATI summary of the three-phase line technologies**

| 132 kV<br>Three-phase AC | ATI     |         |
|--------------------------|---------|---------|
|                          | Minimum | Maximum |
| Pelican                  | 4.3     | 4.9     |
| Twin Hare                | 3.6     | 4.1     |
| Wolf                     | 5.0     | 5.8     |

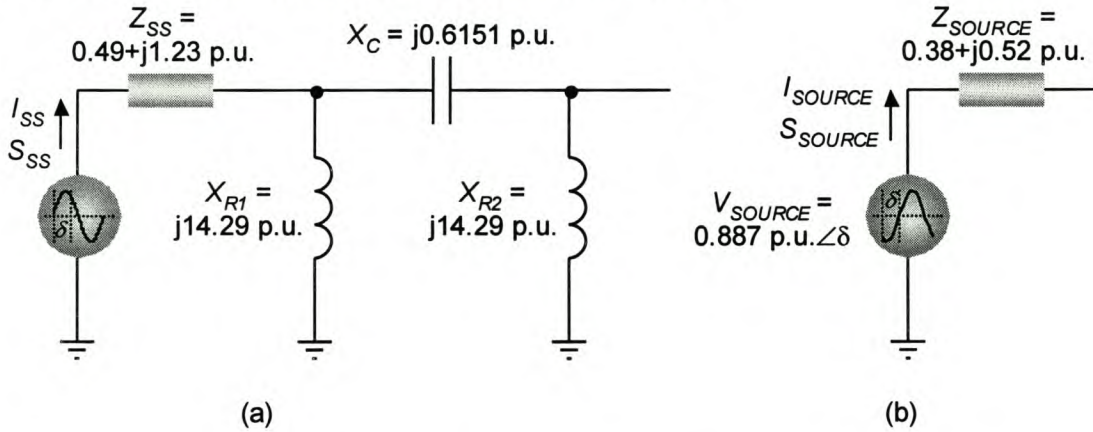
## 4.2.2 Obtaining network parameters

The Caprivi network can connect to the Namibian electricity network at Rundu substation at 132 kV. Nampower supplied the source impedance at the 132 kV and 66 kV busbars at Rundu substation and at the 66 kV busbar at Katima Mulilo distribution station. These values are shown in Table 4-6 below. The high source impedance at the 132 kV Rundu busbar is of particular concern. This high value is due to the long lines that characterise the Namibian electricity network. A switched 7 MVar shunt reactor, with a corresponding reactance of  $j2489.89 \Omega$ , is connected at the 132 kV Rundu busbar to compensate for a voltage rise at Rundu substation during light loading conditions. The connection of another 500 km line with a relatively small point load at this weak point will just worsen the voltage regulation problems, unless some form of compensation is provided to reduce the source impedance.

**Table 4-6: Source impedances**

| Bus           | Voltage [kV] | Impedance [ $\Omega$ ] |
|---------------|--------------|------------------------|
| Rundu         | 132          | $85.23 + j214.34$      |
| Rundu         | 66           | $25.70 + j77.52$       |
| Katima Mulilo | 66           | $58.68 + j143.27$      |

The source impedance is compensated for using a series capacitor and two identical shunt reactors connected on either side of the capacitor as shown in (a) of Fig. 4.10. The purpose of the series capacitor is to reduce the source inductance, while the purpose of the shunt reactors is to control the line voltage on either side of the series capacitor [B17]. The series compensating capacitor was chosen to compensate for 50% of the source reactance, so that  $X_C = j107.17 \Omega$  (450 kVar).  $X_{R1}$  is the 7 MVar switched busbar reactor at Rundu substation. To save costs this reactor is reconfigured as a line reactor on the one side of the series capacitor and a second identical line reactor,  $X_{R2}$ , is added on the other side. All values in Fig. 4.10 are per unit, with base values  $S_{base} = 100$  MVA and  $V_{base} = 132$  kV.



**Fig. 4.10: Compensated source**

The model in (a) of Fig. 4.10 can be simplified by Thévenin-Norton source transformations until the network corresponds to (b), where  $V_{SOURCE}$  and  $Z_{SOURCE}$  are calculated as follows:

$$V_{SOURCE} = \frac{X_{R1} X_{R2} V_{SS}}{(Z_{SS} + jX_{R1}) \left( \frac{jZ_{SS} X_{R1}}{Z_{SS} + jX_{R1}} - jX_C + jX_{R2} \right)} \quad (4-1)$$



$$Z_{SOURCE} = \frac{j \left( \frac{jZ_{SS}X_{R1}}{Z_{SS} + jX_{R1}} - jX_C \right) X_{R2}}{\frac{jZ_{SS}X_{R1}}{Z_{SS} + jX_{R1}} - jX_C + jX_{R2}} \quad (4-2)$$

From section 4.2.1 it was determined that Wolf conductor with a TRANS 980 structure would be the optimum conductor-tower combination for the AC network. The series impedance and shunt admittance was calculated previously by the OLP interface of PSAT for the specified conductor and line structure. These values are then used to calculate the ABCD parameters, which is an exact representation of the line. It would, however, be more convenient to have a circuit representation of the line. The equivalent  $\pi$  circuit model is calculated in the OLP interface to have identical ABCD parameters to what was calculated previously for the specific conductor and line structure.

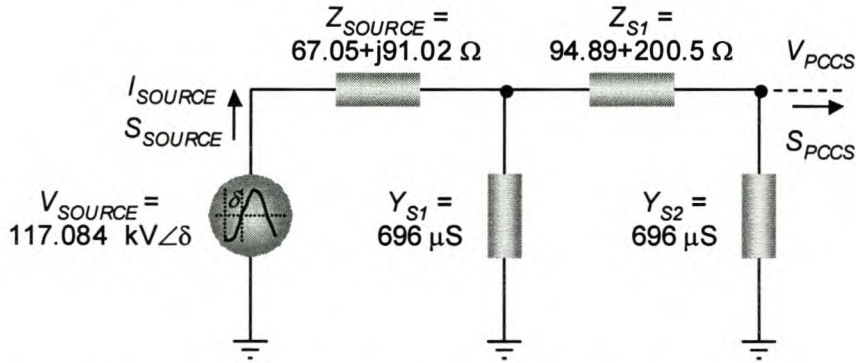


Fig. 4.11: Compensated source impedance and equivalent  $\pi$  circuit model

The compensated source and equivalent  $\pi$  circuit model are shown in Fig. 4.11.  $Z_{S1}$  and  $Y_{S1}$  are the total series impedance and shunt admittance of the line, respectively. This model is implemented in OLP and can be simplified by one or two Thévenin-Norton source transformations. The results,  $V_S$  and  $Z_S$ , are converted to per unit values and exported to the sending-end of PSAT. Equations (4-3) and (4-4) are used to calculate  $V_S$  and  $Z_S$ :

$$V_S = \frac{V_{SOURCE} Z_{S11} Z_{S12}}{(Z_{S11} Z_{SOURCE} + Z_{S1} Z_{SOURCE} + Z_{S1} Z_{S11} + Z_{S12} Z_{SOURCE} + Z_{S12} Z_{S11})} \quad (4-3)$$

$$Z_S = \frac{Z_{S12} (Z_{SOURCE} Z_{S11} + Z_{SOURCE} Z_{S1} + Z_{S12} Z_{S11})}{(Z_{S11} Z_{SOURCE} + Z_{S1} Z_{SOURCE} + Z_{S1} Z_{S11} + Z_{S12} Z_{SOURCE} + Z_{S12} Z_{S11})} \quad (4-4)$$

$$= R_S + jX_S$$

where  $Z_{S11} = Y_{S1}^{-1}$  and  $Z_{S12} = Y_{S2}^{-1}$ .

### 4.2.3 Uncompensated network

The simplified model of the complete network, including the compensated source, the 500 km line and the initial load at Katima Mulilo, is shown in Fig. 4.12. This is a radial network and has only one voltage source, namely Rundu substation. A voltage-regulation problem is anticipated for initial load conditions. Due to the long distance and small point load, there is a rise in the receiving-end voltage, due to the flow of capacitive (line-charging) current through line inductance (Ferranti effect).

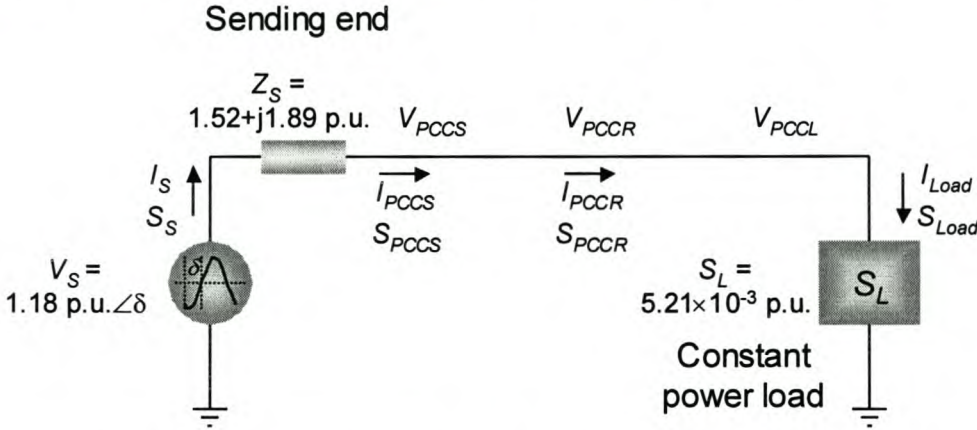
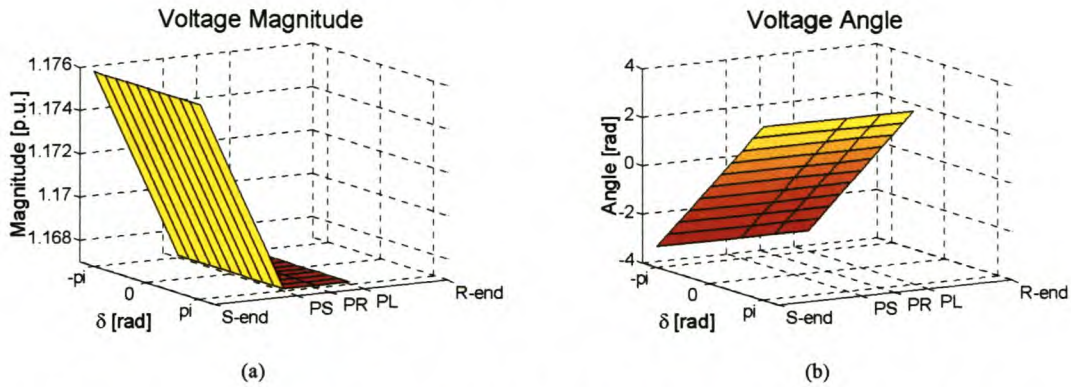


Fig. 4.12: Equivalent circuit diagram for the Katima Mulilo network

The circuit in Fig. 4.12 was analysed in PSAT. The phase angle of  $V_S$  ( $\delta$ , as indicated in Fig. 4.12) was given a fixed value when the network parameters were determined. All other parameters were kept constant while  $\delta$  was varied from  $-\pi$  to  $\pi$  to simulate various voltage angles across the network. A higher  $\delta$  generally results in a higher power transfer across a network, but in practice uncompensated networks are seldom operated at power angles larger than 0.7 rad. The possibility exists that a compensated network can be operated at larger voltage angles because the compensator can increase the first swing stability margin [B3].

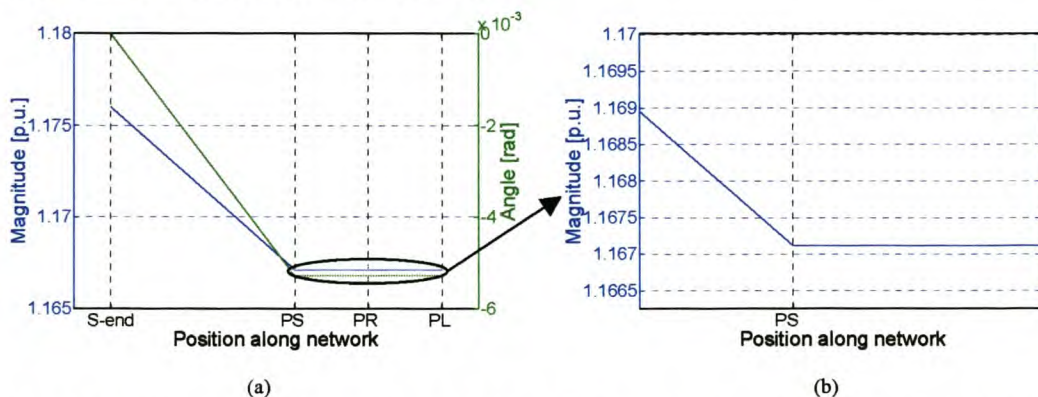
The voltage plots indicating the voltage regulation for this network are shown in Fig. 4.13. The figure shows the voltage regulation across the network of Fig. 4.12. The physical location on the network is plotted on the Y-axis and  $\delta$  is plotted on the X-axis. Therefore, the Z-axis represents the respective voltage magnitude and angle at a specific location in the network, with a certain phase angle across the network. The graphs are plotted for  $-\pi \leq \delta \leq \pi$ .





**Fig. 4.13: Voltage regulation as a function of network position and voltage angle for the uncompensated Katima Mulilo network**

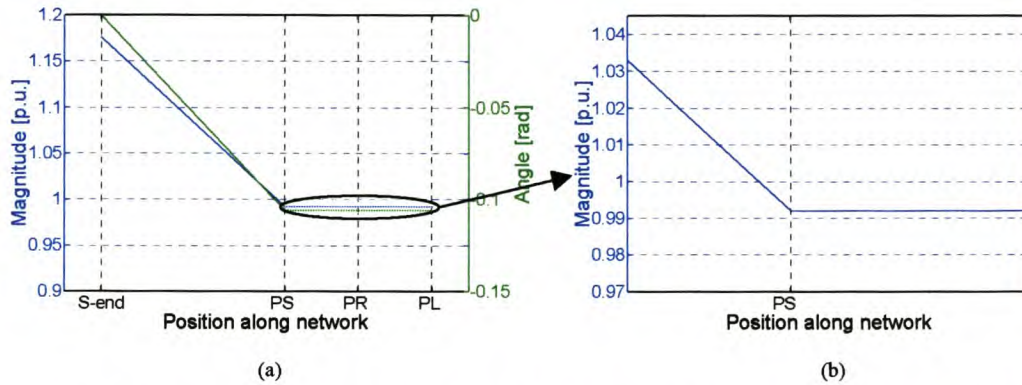
The voltage regulation problem at the point of common coupling (PCC) is clearly visible in the magnitude plot in (a). The sending end, receiving end and the load are situated at the PCC. In all plots the PCC is labelled as follows: PS for PCCS (sending-end side of PCC), PR for PCCR (receiving-end side of PCC) and PL for PCCL (load side of PCC). Due to the constant load at Katima Mulilo and only one voltage source in the network, the voltage magnitude at a specific location in the network remains constant for all  $\delta$ . For no receiving-end source or impedance, the voltage graphs stop at the PCC. The voltage drop across  $Z_S$  is visible between the sending end and PCCS in both the voltage magnitude (a) and angle (b) graphs. For a radial network the relative voltage angle across the network stays constant for all  $\delta$ . The difference in the voltage angles between the sending end and the PCC is constant for all  $\delta$  and is due to the voltage drop across  $Z_S$ . The third dimension therefore contains no information. A cut parallel to the YZ-plane at  $\delta = 0$  reveals the true extend of the voltage regulation problem.



**Fig. 4.14: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and an initial load of  $5.21 \times 10^{-3}$  p.u.**



In Fig. 4.14  $|V_{PCCR}|$  equals 1.167 p.u., which is 0.117 p.u. above NRS 048 standard of 1.05 p.u. for >500 V networks.  $|V_{PCCS}|$  also equals 1.167 p.u. ( $V_{PCCS}$  and  $V_{PCCR}$  are physically at the same point for the uncompensated network), which is not acceptable for loads along the line. In this case no loads are connected along the line, but the line equipment is stressed, which in turn reduces its lifespan. This increases the probability of flashovers and failures. A higher load demand will improve the voltage regulation at the PCC.



**Fig. 4.15: Voltage regulation as a function of network position for the uncompensated Katima Mulilo network and a final load of  $88.54 \times 10^{-3}$  p.u.**

In Fig. 4.15  $|V_{PCCR}|$  and  $|V_{PCCS}|$  equals 0.992 p.u which is within NRS 048 limits. The voltage regulation of the anticipated final load for the Katima Mulilo network, as well as all the points along the line, will be within the required limits. No compensation, therefore, will be required for higher load demands. The negative voltage angle at the PCCR is due to the voltage drop across  $Z_S$  while the sending-end voltage angle,  $\delta$ , is zero.

### Transmission cost of the uncompensated network

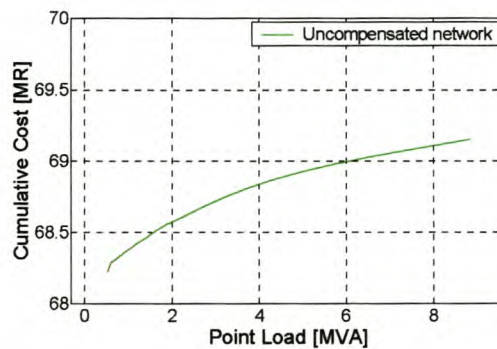
The Caprivi network is supplied from Katima Mulilo distribution station at 66 kV. An 10 MVA 132/66 kV transformer must, therefore, be installed at Katima Mulilo distribution station and the cost included in the LCC of the uncompensated network. The equipment needed to reduce the source impedance at Rundu substation also increases the capital cost of the uncompensated network. These capital costs are summarised in Table 4-7.



**Table 4-7: Additional capital costs incurred for the uncompensated Katima Mulilo network**

| Description                          | Total [R] |
|--------------------------------------|-----------|
| 10 MVA 132/66 kV transformer         | 1 112 000 |
| 450 kVar capacitor bank              | 4 800     |
| Switchgear, civil work and steelwork | 1 500 000 |
| Primary over-voltage protection      | 1 200 000 |
| Control and protection               | 210 000   |
| 7 MVar shunt reactor bank            | 700 000   |

Nampower supplied the costs of the transformer and the shunt reactor bank. The costs of the series capacitor and the equipment to protect the capacitor against fault conditions, as well as other related costs, were supplied by ABB. The present value of LCC ( $PV_{LCC}$ ) for the lifetime of the uncompensated Katima Mulilo network is shown in Fig. 4.16



**Fig. 4.16: LCC of the uncompensated Katima Mulilo network**

A summary of the minimum and maximum present values of LCC over the lifetime of the uncompensated AC network is shown in Table 4-8.  $PV_{LCC}$  included the capital investment cost of the 500 km, 132 kV three-phase Wolf line, the transformer at Katima Mulilo distribution station and the source compensation equipment at Rundu substation. Only the cost of losses in the line ( $PV_{TCL}$ ) was included in  $PV_{LCC}$ , whereas the cost of losses in the transformer and the equipment required for source compensation were not included.

**Table 4-8: Summary of the LCC of the uncompensated Katima Mulilo network**

| Network Configuration | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|-----------------------|-----------------|---------|-----------------|
|                       | Minimum         | Maximum |                 |
| Uncompensated         | 68.23           | 69.16   | 0.93            |

The voltage regulation problems and the high cost of the equipment needed to protect the series capacitor during fault conditions raises suspicions about the effectiveness of an AC interconnection in terms of cost and technical performance. In the sections to follow shunt, series and series-shunt compensator topologies will be investigated and analysed as solutions to the voltage regulation problem of the AC network for small loads. DC transmission, as an alternative to an AC interconnection, will also be investigated in order to determine the optimum solution for this network.

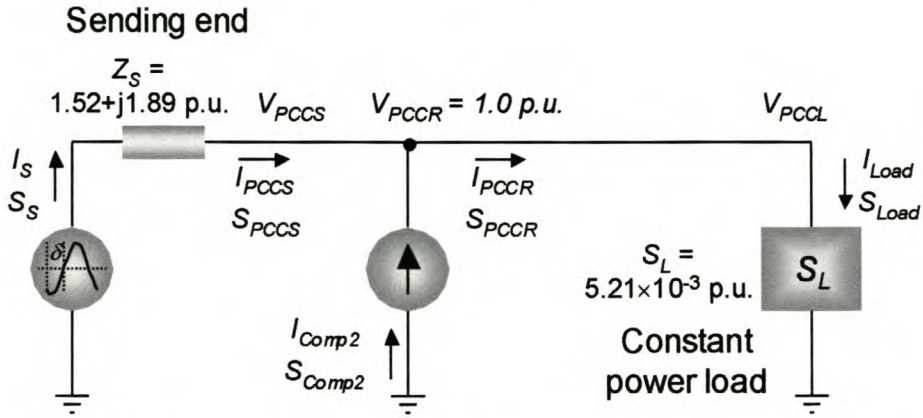


### 4.3 SHUNT COMPENSATION

Katima Mulilo has a 3.75 MVA diesel-fired power station, which consists of six generators with a capacity of 0.5 MW and 0.375 MVar each. These generators only turn in during outages. Using the generators as synchronous condensers may be a solution to the voltage regulation problem for light loading conditions. The generators will be under-excited in order to absorb reactive power. Used in this manner, the generators inject no active power into the network and therefore do not expend any fuel (no energy storage required). With the generators connected directly to the AC network, reaction time is in the order of cycles and continuously variable compensation is achieved.

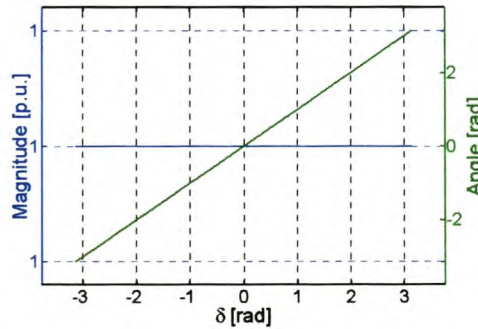
An alternative solution is to install shunt reactors, which will absorb purely reactive power. The shunt reactors are used to provide inductive reactance to control system voltage and to compensate for the effects of the high charging current of long lines. For light load conditions this charging current can produce more leading reactive power than the system can absorb without risk of instability or excessively high voltages at the line terminals [A13]. Apart from being switched, these devices are incapable of continuous variable compensation. An intelligent controller, which monitors the voltage waveform, can be used to switch the reactors when the network voltage exceeds the NRS 048 specifications. Such a controller can reduce switching cycles to milliseconds. Switching transients introduced by the reactor switching are also a major drawback of this method.

Fig. 4.17 is the circuit diagram of the shunt-compensated network.  $|V_{PCCR}|$  is specified as 1.0 p.u. to adhere to NRS 048 specification. Note that the voltage angle at the sending end is now taken as the reference angle (0 rad), while the voltage angle at the PCC is varied from  $-\pi \leq \delta \leq \pi$ . The network parameters are known from Fig. 4.12.



**Fig. 4.17: Katima Mulilo network with a shunt compensator for light loads**

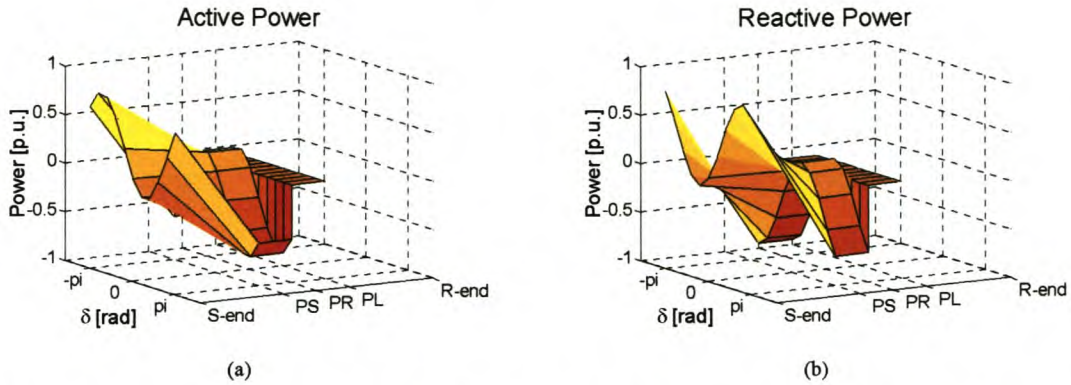
The voltage diagram generated with PSAT at the PCCR and PCCS, which are the same points for shunt compensation, is shown in Fig. 4.18.  $|V_{PCCR}|$  is lowered to 1.0 p.u. for all  $\delta$ , which is how the shunt compensator is expected to work. Fig. 4.18 shows that the voltage regulation for light loads, as well as all the points along the line, will be within the required limits. As for the uncompensated network, the difference in voltage angles is constant for all  $\delta$  and is due to the voltage drop across  $Z_S$ .



**Fig. 4.18: Voltage plots at the PCC**

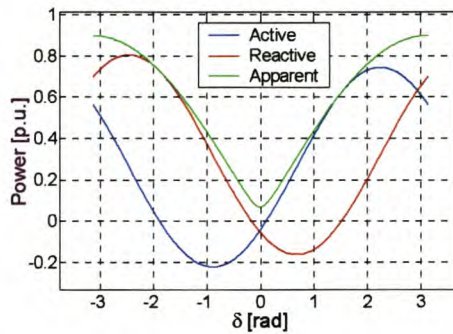
Although phase-shifted by the losses in the network, the reactive power demand at the sending end is a typical co-sinusoidal function of  $\delta$ , while the active power demand is a sinusoidal function of  $\delta$ , as shown in Fig. 4.19. The chosen direction of power flow is shown in Fig. 4.17.





**Fig. 4.19: Active and reactive power dissipated across the shunt-compensated Katima Mulilo network**

The shunt compensator, by adding its active and reactive power to the network, creates the step in active and reactive power at the PCCR. The step at the PCCL in the active and reactive power graphs is the active and reactive power dissipated by the load at Katima Mulilo distribution station. PSAT shows the compensator power as a separate two-dimensional graph, shown in Fig. 4.20.



**Fig. 4.20: Power plots for the shunt compensator**

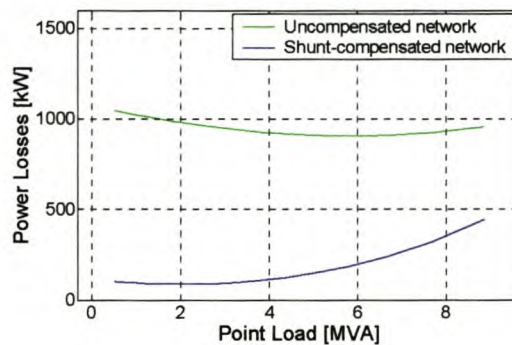
The compensator has a minimum rating of 0.0683 p.u. (6.83 MVA). The active and reactive power absorbed are 0.0404 p.u. (4.04 MW) and 0.0551 p.u. (5.51 MVar), respectively. This minimum rating is achieved at  $\delta = 0$  rad. This is well above the capability of the generators at the diesel-fired power station. Also, minimum compensation requires a power factor of -0.59, whereas the generators can only operate down to a power factor of 0.8.

A purely reactive compensator can be implemented when the active power is zero. This occurs at  $\delta = -1.88$  rad and  $\delta = 0.1$  rad. The latter is chosen as this delivers the lowest compensator rating of 0.086 p.u. A switched reactor bank capable of absorbing 8.6 MVar at the PCCR is therefore required for this control strategy.

Alternatively, the generators at Katima Mulilo power station can be modified to operate as synchronous condensers. The cost involved will be that of modifying the generators to operate as synchronous condensers. Switched shunt reactor banks can be installed to absorb the bulk of reactive power, whereas the generators can be used to rapidly compensate the balance of the required reactive power. The trade-off is the reduction in the capital investment of providing the equipment to absorb the reactive power.

### Transmission losses of the purely reactive shunt-compensated network

Fig. 4.21 shows the effect of purely reactive shunt compensation on the power losses of the uncompensated network. For the uncompensated network high line charging currents produce more leading reactive power than the system can absorb, causing high power losses and excessively high voltages at the line terminals. The effect of purely reactive shunt compensation is to absorb the balance of the leading reactive power in the network, reducing power losses and overvoltages in the network.



**Fig. 4.21: Effect of purely reactive shunt compensation on the power losses of the uncompensated network**

A shunt reactor bank of 7 MVar is installed to absorb the bulk of reactive power. The generators at Katima Mulilo power station are reconfigured to operate as synchronous condensers to absorb the rest of the reactive power (maximum of 1.6 MVar). Table 4-9 provides a summary of the total power losses for uncompensated and shunt compensated network in Fig. 4.21. The minimum losses for the shunt-compensated network are 90% less than the minimum losses for the uncompensated network, whereas the maximum losses for the shunt-compensated network are 58% less. In Fig. 4.21 the additional power losses in the shunt reactors banks and synchronous



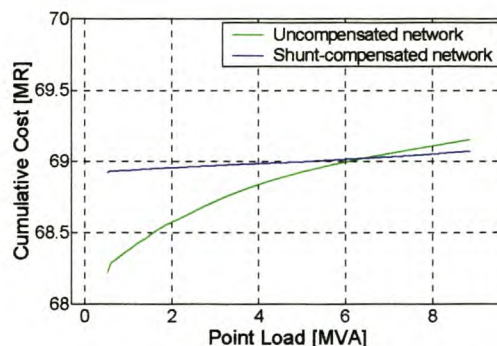
condensers were not included in the total power losses for the shunt-compensated network.

**Table 4-9: Summary of total power losses for uncompensated and shunt-compensated Katima Mulilo network**

| Network Configuration | Minimum           |              | Maximum           |              |
|-----------------------|-------------------|--------------|-------------------|--------------|
|                       | $P_{phase\ loss}$ | $P_{Losses}$ | $P_{phase\ loss}$ | $P_{Losses}$ |
|                       | [kW]              | [kW]         | [kW]              | [kW]         |
| Uncompensated         | 303               | 910          | 350               | 1 050        |
| Shunt compensated     | 30                | 90           | 148               | 444          |

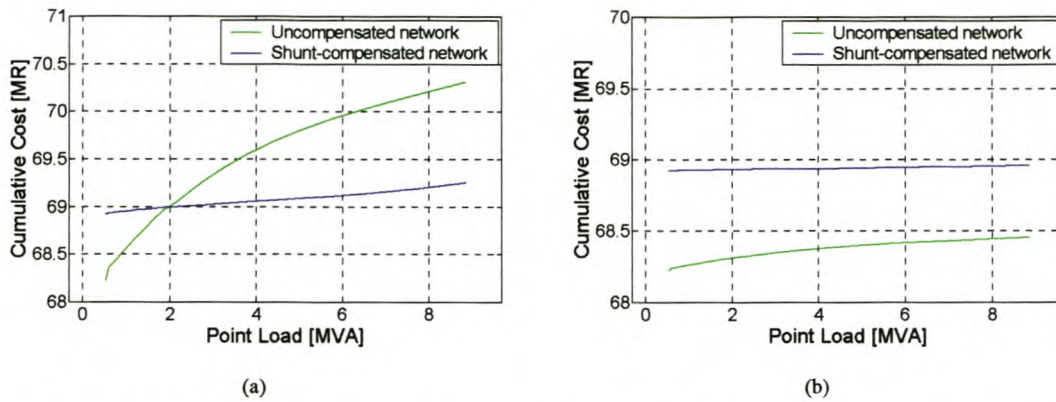
### Transmission cost of the purely reactive shunt-compensated network

Using shunt reactors and static condensers to compensate for overvoltages imply increased capital cost incurred. The cost of losses must therefore be used as outflows of capital in determination of the minimum present value of LCC. Fig. 4.22 shows the present value of LCC for the uncompensated and shunt-compensated networks for an ALF of 0.4. Nampower supplied the cost of the 7 MVar shunt reactor banks (Table 4-7). The cost of reconfiguring the generators as synchronous condensers is much less than the capital cost of the shunt reactor banks and was, therefore, not included.



**Fig. 4.22: LCC of the uncompensated and shunt-compensated Katima Mulilo network (ALF = 0.4)**

From an economic point of view it can be concluded from Fig. 4.22 that only in the 18<sup>th</sup> year does the shunt-compensated network become more cost effective than the uncompensated network. Thus, only after 18 years are the savings in the cost of losses with the shunt compensated network great enough to pay for the shunt reactors.



**Fig. 4.23: LCC of the uncompensated and shunt-compensated Katima Mulilo network for an ALF of (a) 0.6 and (b) 0.2**

The effect of increasing the ALF to 0.6 is shown in Fig. 4.23 (a). Increasing the load factor will reduce the number of years to pay back the initial capital investment of shunt compensation. In this case the payback period will be reduced from 18 years to 10 years. Similarly by reducing the ALF to 0.2 will increase the payback period. This is shown in Fig. 4.23 (b). Table 4-10 provides a summary of the minimum and maximum present values of the LCC over the lifetime of the uncompensated and shunt-compensated AC networks for different values of ALF.

**Table 4-10: Summary of the LCC of the uncompensated and shunt-compensated Katima Mulilo network for different values of ALF**

| Network Configuration | ALF | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|-----------------------|-----|-----------------|---------|-----------------|
|                       |     | Minimum         | Maximum |                 |
| Uncompensated         | 0.2 | 68.23           | 68.46   | 0.23            |
| Uncompensated         | 0.4 | 68.23           | 69.16   | 0.93            |
| Uncompensated         | 0.6 | 68.23           | 70.32   | 2.09            |
| Shunt compensated     | 0.2 | 68.93           | 68.96   | 0.04            |
| Shunt compensated     | 0.4 | 68.93           | 69.07   | 0.15            |
| Shunt compensated     | 0.6 | 68.93           | 69.26   | 0.33            |

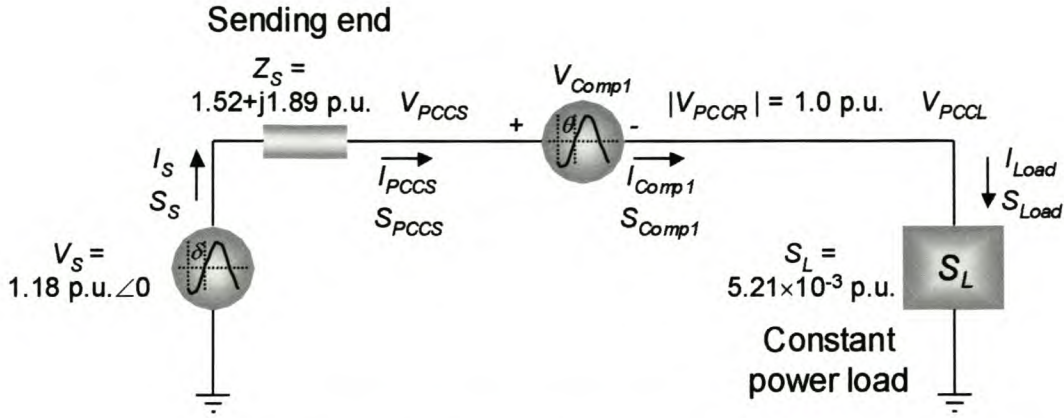


## 4.4 SERIES COMPENSATION

Series compensators are widely recognised as voltage regulators. A series controller could be a variable impedance, such as capacitor, reactor, etc., or a power electronics-based variable source of main frequency, subsynchronous and harmonic frequencies (or a combination) to serve the desired need [B3]. In principle all series compensators inject a voltage in series with the line. Even variable impedance multiplied by the current flow through it represents an injected series voltage in the line. The drawback of series compensation is that full load current flows through the compensator, which can be problematic under fault conditions.

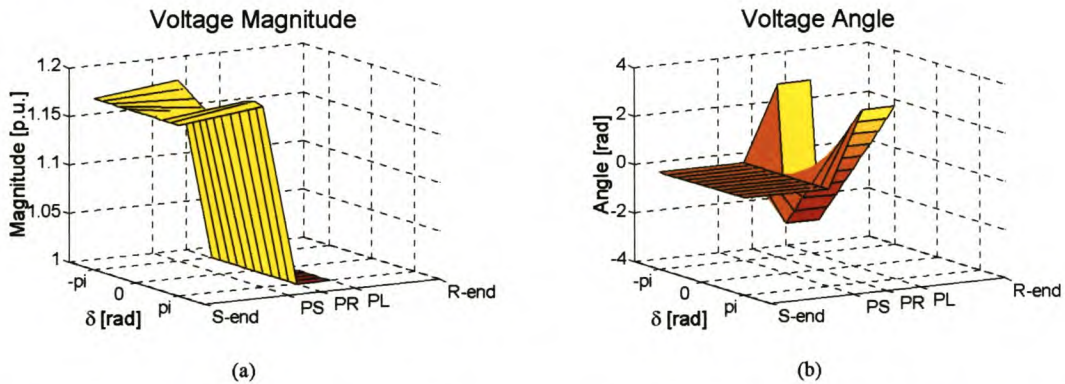
The standby generators at Katima Mulilo power station cannot be used for series compensation by itself. It can, however, be used as an energy source connected through a regulator to the DC bus of a transistor-based series device. Depending on the rating of the device, this can be a very costly option. Series capacitors are a viable alternative to shunt compensation due to their relatively low cost. Depending on the controller, series capacitors are also limited by their switching times, as discussed in section 4.3. Since series capacitors compensate for the line impedance, they are self-regulating in terms of load variations. Switching is, however, often required for stability reasons. Increasing in cost and complexity, a Thyristor-Controlled Series Capacitor (TCSC) or a transistor-based device, namely a Static Synchronous Series Compensator (SSSC), can also be used as a voltage regulator [B3]. The TCSC can supply only reactive power with a reaction time of half a cycle, while the SSSC, if equipped with energy storage, can supply both active and reactive power in less than a millisecond [C1].

Fig. 4.24 is the circuit diagram of the series-compensated network.  $|V_{PCCR}|$  is specified as 1.0 p.u. to adhere to NRS 048 specification. The network parameters are known from Fig. 4.12.



**Fig. 4.24: Katima Mulilo network with a series compensator**

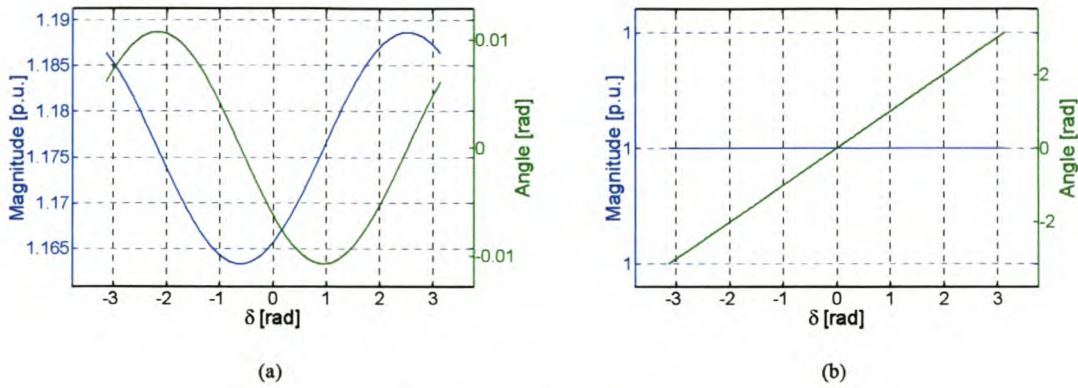
The magnitude and angle across the series-compensated Katima Mulilo network are shown in Fig. 4.25. The voltage magnitude and angle injected by the series compensator varies as the voltage angle at the PCC is varied from  $-\pi \leq \delta \leq \pi$  (between PCCS and PCCR). For different values of  $\delta$ , the series compensator can be modelled as variable impedance connected in series with the line. By varying the line impedance with  $\delta$ , the voltage drop across the line will not be constant for all values of  $\delta$ , as shown in Fig. 4.25.



**Fig. 4.25: Voltage magnitude and angle across the series-compensated Katima Mulilo network**

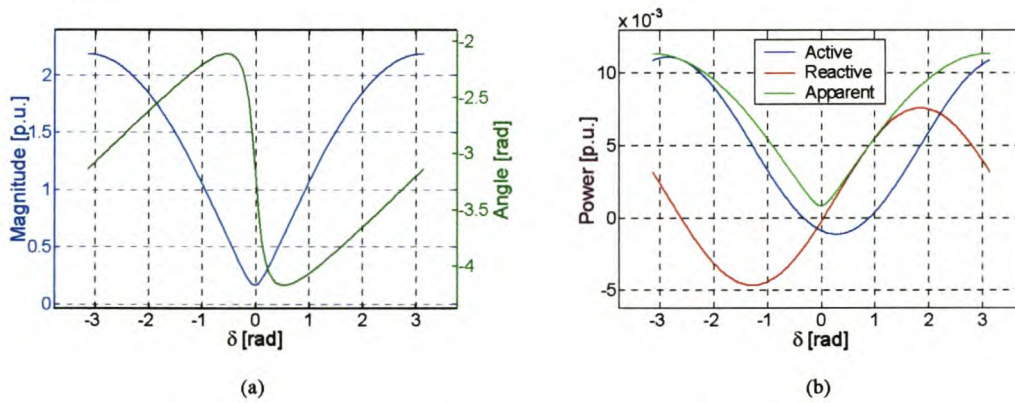
The voltage diagrams before and after the compensator at PCCS and PCCR are shown in Fig. 4.26 for an initial load of  $5.21 \times 10^{-3}$  p.u. The voltage regulation at PCCR is as desired, but varies between 1.16 and 1.19 p.u. at PCCS, which is acceptable as long as no loads are connected along the line and the line equipment is rated for overvoltages at PCCS.





**Fig. 4.26: Voltage plots at (a) PCCS and (b) PCCR**

The true voltage added by the series compensator, as well as the power injected by the compensator, is shown Fig. 4.27. Although the series compensator must carry full load current, the voltage across the compensator is quite small (0.1658 p.u. for a minimum rating series compensator and 0.3779 p.u. for a purely reactive series compensator).



**Fig. 4.27: (a) Voltage and (b) power plots for the series compensator**

The compensator has a minimum rating of  $8.636 \times 10^{-4}$  p.u. or 86.36 kVA with a power factor of  $-0.97$  (83.77 kW and 20.99 kVar). This minimum rating is achieved at  $\delta = 0$  rad. Compared to the minimum shunt voltage regulator, the series device rating is 97.93% less active power and 99.62% less apparent power. A SSSC equipped with energy storage can be used to supply both active and reactive power. As discussed previously, the generators at Katima Mulilo power station can be used as an energy source for this control scheme.

A purely reactive compensator can be implemented when the active power is zero. This occurs at  $\delta = -0.32$  and  $\delta = 0.8875$  rad. The former is chosen as this delivers the lowest compensator rating of  $-1.97 \times 10^{-3}$  p.u. A series capacitor or a more complex

device, such as a TCSC, with a rating of 197 kVar is required for this control strategy. Compared to purely reactive shunt compensation, the series device rating is 97.71% less reactive power, but requires additional bypass switches to prevent network fault currents from flowing through the compensator.

Transmission cost of the purely reactive series-compensated network

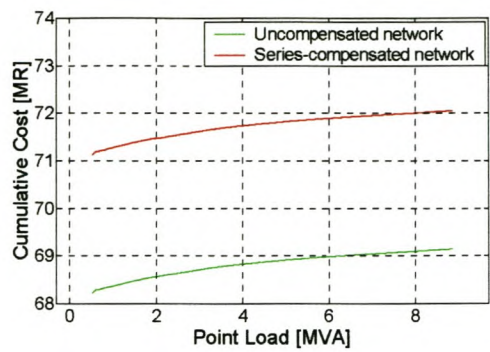
Because this is a series network the line current magnitude and angle stays constant throughout the network. The power losses of the series-compensated network will therefore be the same as for the uncompensated network discussed previously in section 4.2.3. The capital cost of the series capacitor and the equipment needed to bypass the capacitor during fault conditions just increases the LCC of the already expensive uncompensated AC network. For the series-compensated case the cost of losses cannot be used as outflows of capital in determination of the minimum present value of LCC, because the power losses are the same for both the uncompensated and the series-compensated network. The cost of the series capacitor and additional protection and control equipment were supplied by ABB and is shown in Table 4-11.

**Table 4-11: Additional capital costs incurred for the series-compensated Katima Mulilo network**

| Description                          | Total [R] |
|--------------------------------------|-----------|
| 197 kVar capacitor bank              | 2 100     |
| Switchgear, civil work and steelwork | 1 500 000 |
| Primary over-voltage protection      | 1 200 000 |
| Control and protection               | 210 000   |

Fig. 4.28 shows the present value of LCC for the lifetime of the uncompensated and series-compensated Katima Mulilo network and the results are summarised in Table 4-12.





**Fig. 4.28: LCC of the uncompensated and series-compensated Katima Mulilo network**

The increased capital cost incurred with series compensation is clearly visible in Fig. 4.28. Due to the high cost of the equipment to protect the series capacitor against fault conditions, the purely reactive shunt compensator will be a more cost effective solution than the series compensator.

**Table 4-12: Summary of the LCC of the uncompensated and series-compensated Katima Mulilo network**

| Network Configuration | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|-----------------------|-----------------|---------|-----------------|
|                       | Minimum         | Maximum |                 |
| Uncompensated         | 68.23           | 69.16   | 0.93            |
| Series compensated    | 71.14           | 72.07   | 0.93            |

## 4.5 SERIES-SHUNT COMPENSATION

As its name indicates, the series-shunt compensator is a combination of the two devices. The series and shunt devices are fed from a common DC bus and, therefore, can exchange active power between them. The series-shunt device, therefore, has no need for an energy-storage device. With two devices, which can do both active and reactive power compensation, a potentially powerful and highly effective compensator is realized. Such a device not only has full four-quadrant operating capability in the complex  $P + jQ$  plane, but also internal Var generation capability to support self-sufficiently the reactive power demand in any quadrant [B3].

With the network parameters from Fig. 4.12, the circuit diagram of the series-shunt compensated network is drawn in Fig. 4.29. Both the magnitude and the angle of  $V_{PCCR}$  can be controlled by the series part of the compensator. For consistency,  $|V_{PCCR}|$  is specified to 1.0 p.u.  $\theta_{PCCR}$  is also needed and specified as 0.1 rad. The load does not have a constant voltage or constant current characteristic, so  $\theta_{PCCR}$  makes no difference to the load [C1]. The shunt compensator controls the power factor at the PCCS with reactive power injection. It is taken as unity to limit reactive power demand from Rundu substation at the sending end.

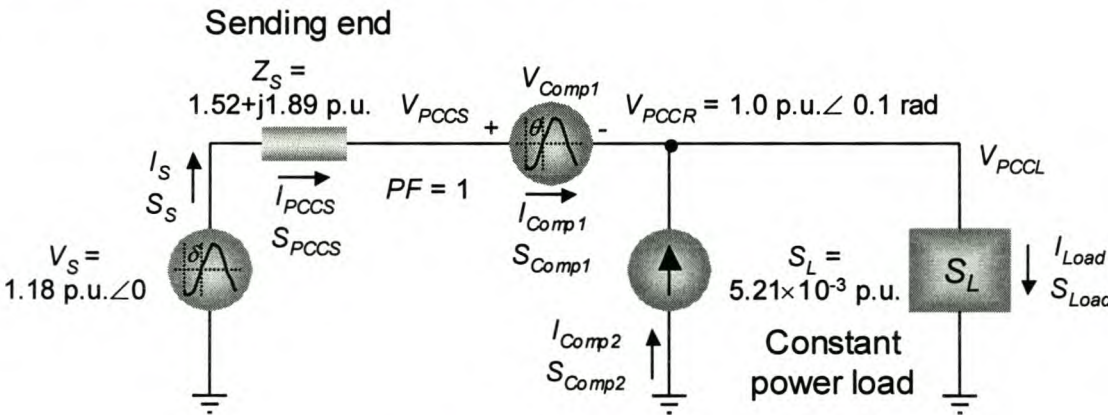
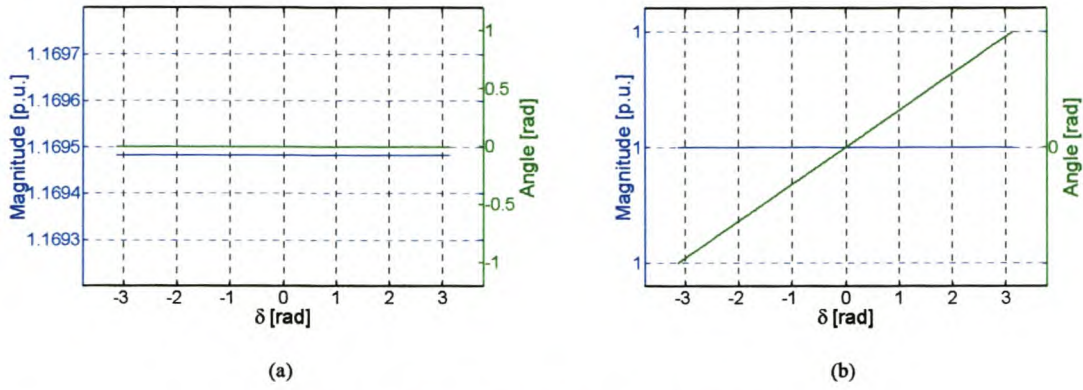


Fig. 4.29: Katima Mulilo network with a series-shunt compensator

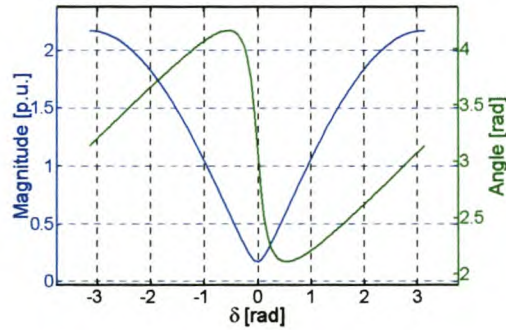
The voltage diagrams generated by PSAT at the PCCS and PCCR are shown in Fig. 4.30 for an initial load of  $5.21 \times 10^{-3}$  p.u. The voltage regulation at the PCCR is the desired 1.0 p.u., but 1.1695 p.u. at the PCCS, which is well above NRS 048 specifications. No loads are connected along the line, but the line equipment must be rated for overvoltages at initial load conditions.





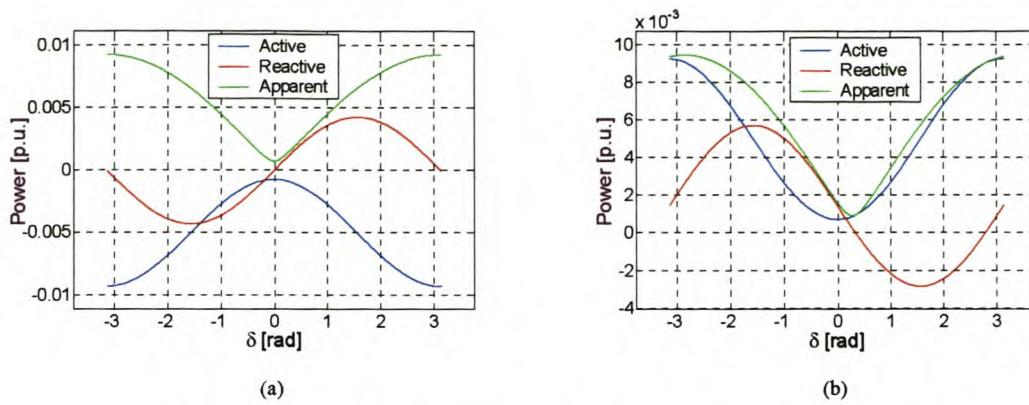
**Fig. 4.30: Voltage plots at (a) PCCS and (b) PCCR**

The true voltage injected by the series compensator to regulate  $|V_{PCCR}|$  is shown in Fig. 4.31. The compensator will have a maximum voltage rating of 2.17 p.u. at  $\delta = 3.14$ . Although the possibility exists that the compensator can increase the first swing stability margin, a compensated network will never be operated at such a high voltage angle. If a maximum voltage angle of 0.7 rad is assumed across the network, the maximum voltage rating will reduce to 0.77 p.u. The compensator will have a minimum rating of 0.17 at  $\delta = 0$ .



**Fig. 4.31: True voltage injected by the series compensator**

The active and reactive power delivery for the separate series and shunt compensators are shown in Fig. 4.32. The active power of the series and shunt devices are the opposite of one another, because the shunt compensator absorbs the active power required by the series compensator. This power is then transferred via the DC bus. DC bus losses are ignored; therefore, the series device injects all active power absorbed by the shunt device back into the network. The reactive power ratings differ because the series device regulates  $V_{PCCR}$ , while the shunt device regulates the power factor at PCCS.



**Fig. 4.32: Power requirements of the (a) series and (b) shunt devices**

The maximum apparent power delivered by the series compensator is  $3.282 \times 10^{-3}$  p.u. (328.2 kVA) at  $\delta=0.7$ . The maximum apparent power absorbed by the shunt compensator is  $2.191 \times 10^{-3}$  p.u. (219.1 kVA). The active power delivered and absorbed per compensator is  $1.75 \times 10^{-3}$  p.u. (175 kW) at this point. It can therefore be reasoned that the maximum rating of the series-shunt compensator is 547.3 kVA (350 kW).

The minimum apparent power delivered by the series compensator is  $7.253 \times 10^{-4}$  p.u. (72.53 kVA) at  $\delta=0$ . The active power delivered and absorbed per compensator is  $7.247 \times 10^{-4}$  (72.47 kW) at this point. The minimum apparent power absorbed by the shunt compensator is  $9.407 \times 10^{-4}$  p.u. (94.07 kVA) at  $\delta=0.2513$ , and the active power delivered and absorbed per compensator is  $8.663 \times 10^{-4}$  (86.63 kW).

Although the power rating of the series-shunt compensator is much less than the shunt device rating, the implementation of a series-shunt compensator for the Katima Mulilo network is technically impractical, due to the high voltage ratings of the series and shunt devices (102 kV and 132 kV). Such a device is economically not feasible and nor is it commercially available.



## 4.6 IN-LINE COMPENSATION – THE DC ALTERNATIVE

Medium-voltage direct current (MVDC) transmission can be used to bridge greater distances with low voltage and low power [A7]. MVDC can be considered as in-line compensation in the sense that it consists of a DC link with a voltage source converter (VSC) connected at either end. The advantages of using a VSC (with a series connection of IGBTs) for DC transmission include the following [A14]-[A17]:

- Independent control of active and reactive power
- Operation against isolated AC networks with no generation of their own
- Can operate at any short-circuit ratio
- Limited need of filters
- No converter transformers
- Can connect isolated generators to the grid or to other isolated loads.

For DC transmission the compensator at the sending end absorbs active and reactive power and transfers the active power across a common DC link to the receiving-end compensator. It, therefore, has no need for an energy-storage device. If the compensators are based on VSC technology, both active and reactive power can be transmitted in any direction. The reactive power capabilities of the converter can be used to control the AC voltages of the networks connected to the converter stations. With this type of operation the converter can be connected into an extremely weak AC network. The AC network can supply the reactive power, but capacitor banks are often used to supply this demand [A18]. The receiving-end compensator transmits the active power received over the DC link to the connected AC network. Since in a VSC the current can be forced to commute, there is no need for a network to commute against. In DC transmission it could then be very advantageous to use the VSC technology to supply passive or isolated networks such as Katima Mulilo.

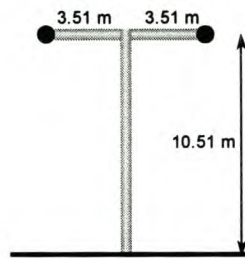
### 4.6.1 *DC line technology optimisation*

The DC interface of TESAT is used to evaluate three different Nampower conductor choices in order to select the optimum conductor for a 500 km,  $\pm 50$  kV bipolar DC line (shown in Fig. 4.1) between Rundu substation and Katima Mulilo distribution

station. The following three ACSR conductors were evaluated with a standard 66 kV Nampower tower:

- Rabbit conductor
- Hare conductor
- Wolf conductor.

The line structure used by Nampower for the above-mentioned conductors are shown in Fig. 4.33. The data for the different conductors were sourced from ACSR property tables and are included in Addendum A.1.



**Fig. 4.33: 66 kV Nampower tower (Kamerad) [C18]**

The DC resistance values at 40°C were calculated from the DC resistance values at 20°C. The resistance values for the different conductors are given in Table 4-13, as well as the deterministic ampacity values and the estimated cost of these lines.

**Table 4-13: ±50 kV bipolar DC line data**

| Conductor | Tower   | $r_{dc} (40^{\circ}\text{C})$<br>[Ω/km] | Ampacity<br>[A] | Line Cost<br>[R/km] |
|-----------|---------|---|-----------------|---------------------|
| Rabbit    | Kamerad | 0.5863                                  | 196             | 68 500              |
| Hare      | Kamerad | 0.2953                                  | 297             | 75 900              |
| Wolf      | Kamerad | 0.1975                                  | 383             | 87 700              |

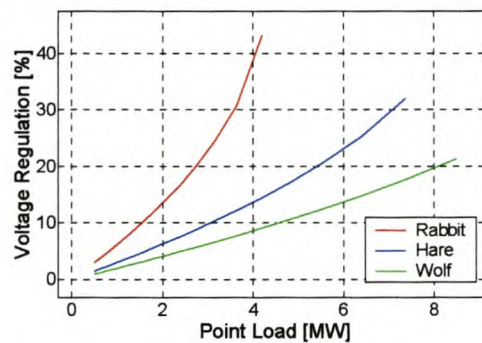
The data in Table 4-13 are entered in the DC interface of TESAT. A base value of 100 MW is used for  $S_{base}$  and 50 kV for  $V_{base}$ . The same values are used for interest rate, generation cost and ALF as for the AC line technologies. The additional cost of the converter stations, one at either end of the line, must be entered by the user, as well as the losses in each station as a percentage of the transmitted power. An estimated cost of R1.8 million per MW for the converter stations was used (R3 600 per MW-km), with the losses in each station estimated as 3% of the transmitted power. Details of the estimated converter station costs are included in



Addendum A.2. The cost per kilometre for the different DC lines was calculated in the same way as for the AC line technologies.

### Voltage regulation analysis

From the TESAT analysis it was determined that the voltage regulation for the Rabbit and Hare conductors exceeds 40% for certain loads and therefore cannot be used to transmit the full load for the lifetime of the network (20 years). The voltage regulation of the  $\pm 50$  kV bipolar Rabbit, Hare and Wolf lines is shown in Fig. 4.34 and the results are summarised in Table 4-14.



**Fig. 4.34: Voltage regulation of the DC line technologies as a function of load growth**

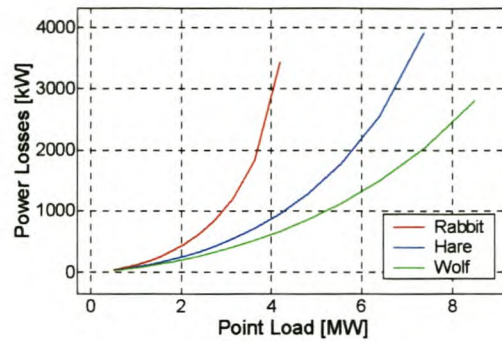
Fig. 4.34 shows that the Rabbit conductor will be able to transmit full load until the 15<sup>th</sup> year, while the Hare conductor will be able to transmit full load until the 19<sup>th</sup> year. Only the Wolf conductor will be able to transmit the full load for the lifetime of the line.

**Table 4-14: Summary of voltage regulation performance**

| $\pm 50$ kV<br>Bipolar DC | % Voltage Regulation |         |
|---------------------------|----------------------|---------|
|                           | Minimum              | Maximum |
| Rabbit                    | 3.0                  | 43.2    |
| Hare                      | 1.5                  | 32.1    |
| Wolf                      | 1.0                  | 21.3    |

### Transmission losses

The power losses in kW for the different bipolar DC lines are shown in Fig. 4.35 and the results are summarised in Table 4-15. These values include the power losses in the converter stations.



**Fig. 4.35: Power losses of the DC line technologies as a function of load growth**

The minimum values of power loss in each of the conductors include a 30 kW loss component for the two converter stations. The maximum value of converter losses for Rabbit and Hare conductors amounts to 251 kW and 443 kW respectively. The Wolf conductor will be the only conductor capable of transmitting the full load for the lifetime of the network, with maximum converter losses equal to 510 kW. The loss per pole ( $P_{pole\ loss}$ ) for each of these lines can be calculated as follows:

$$P_{Losses} = 2 \cdot (P_{support\ loss} + P_{pole\ loss})$$

$$\Rightarrow P_{pole\ loss} = \frac{1}{2} P_{Losses} - P_{support\ loss} \quad (4-5)$$

The results of equation (4-5) are shown in Table 4-15 for each of the bipolar DC lines.

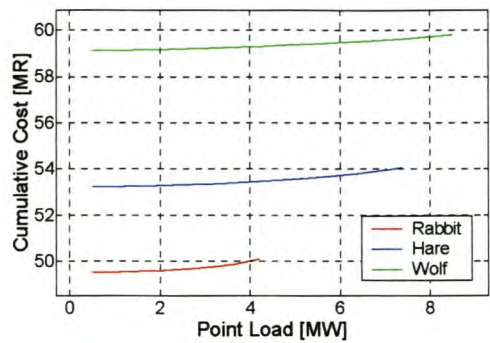
**Table 4-15: Summary of power losses**

| ±50 kV DC Line<br>Technology | Minimum                  |                      | Maximum                  |                      |
|------------------------------|--------------------------|----------------------|--------------------------|----------------------|
|                              | $P_{pole\ loss}$<br>[kW] | $P_{Losses}$<br>[kW] | $P_{pole\ loss}$<br>[kW] | $P_{Losses}$<br>[kW] |
| Rabbit                       | 8                        | 46                   | 1 595                    | 3 440                |
| Hare                         | 4                        | 38                   | 1 741                    | 3 926                |
| Wolf                         | 3                        | 35                   | 1 153                    | 2 816                |

## Transmission costs

Fig. 4.36 shows the present value of LCC for the different bipolar DC lines and the results are summarised in Table 4-16. The capital cost and the cost of losses of the two converter stations are also included in the LCC.





**Fig. 4.36: LCC for the DC line technologies**

From Fig. 4.36 it can be concluded that, although the Rabbit and Hare conductors are the most cost-effective choice to serve the intended load for most of the network lifetime, only the Wolf conductor will be able to transmit the full load for the lifetime of the network. The values of  $PV_{TCL}$  in Table 4-16 include both the cost of losses in the line and the two converter stations for the lifetime of the line.

**Table 4-16: Summary of LCC and cost of losses**

| ±50 kV DC Line<br>Technology | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|------------------------------|-----------------|---------|-----------------|
|                              | Minimum         | Maximum |                 |
| Rabbit                       | 49.55           | 50.11   | 0.56            |
| Hare                         | 53.25           | 54.05   | 0.80            |
| Wolf                         | 59.15           | 59.83   | 0.68            |

Appropriate technology index (ATI)

The KPIs to consider for DC line technologies are LCC, capacity and voltage regulation. As a result of the long distance and the low power requirements of the load, the DC voltage will be relatively high and the current low. Consequently, the capacity KPI of the line is ignored. For DC transmission the transistor-based inverter at the load end can cater for a voltage drop of more than 30%. The optimum line technology for DC transmission has to pursue a low-cost performance, which would economically justify low-power, long-distance transmission. The weighting factors for LCC and voltage regulation are, therefore, chosen as 80% and 20%, respectively.

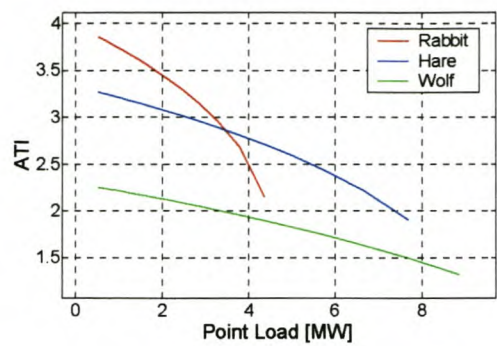


Fig. 4.37: ATI of the bipolar line technologies

The ATI for the bipolar line technologies is shown in Fig. 4.37 and the results are summarised in Table 4-17. Rabbit conductor would be the optimum conductor to string the line with for the first 13 years; after that Hare conductor would be the optimum choice until the 19<sup>th</sup> year. Only after 19 years does the Wolf conductor become the most effective choice in terms of cost and performance. However, owing to the capital cost incurred with conductor stringing, the Wolf conductor will be chosen as the optimum conductor for the network lifetime.

Table 4-17: ATI summary of the bipolar line technologies

| ±50 kV<br>Bipolar DC | ATI     |         |
|----------------------|---------|---------|
|                      | Minimum | Maximum |
| Rabbit               | 2.2     | 3.9     |
| Hare                 | 1.9     | 3.3     |
| Wolf                 | 1.3     | 2.3     |

4.6.2 Simplified network

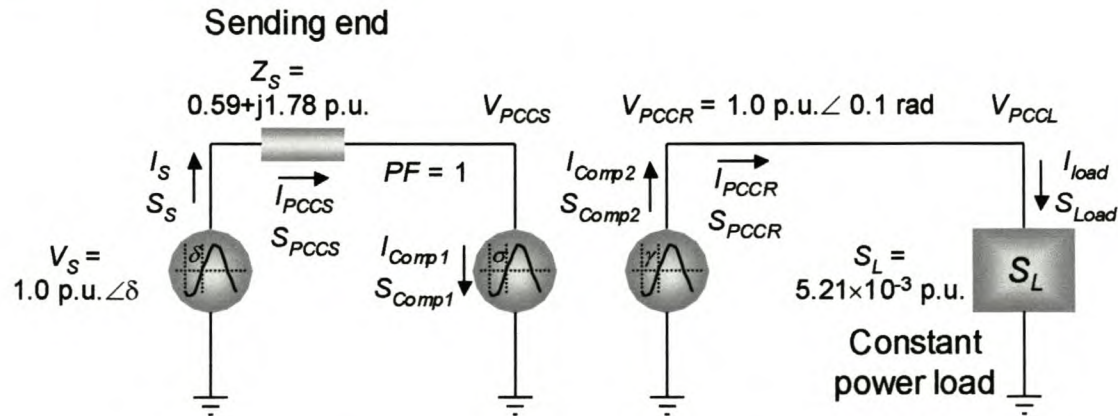
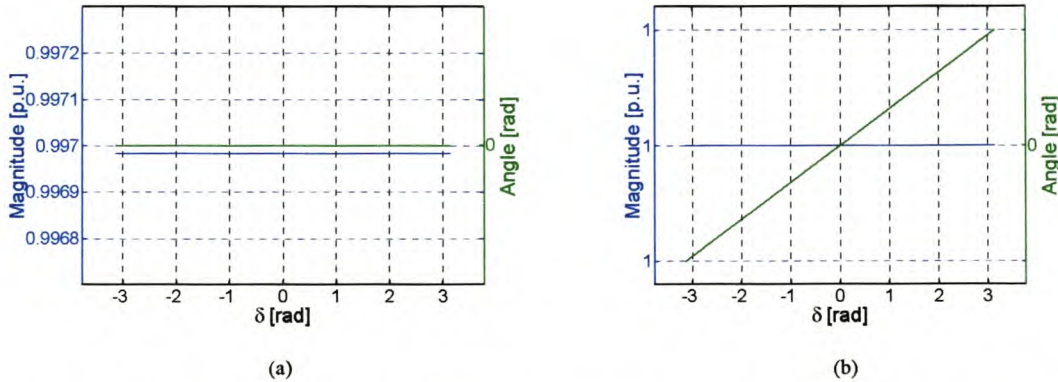


Fig. 4.38: Katima Mulilo network with an in-line compensator



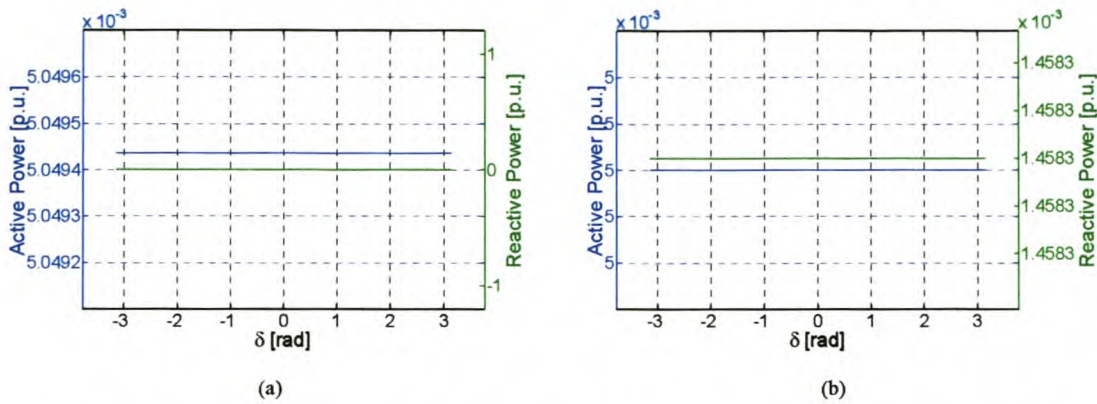
The two compensators in Fig. 4.38, are connected via a 500 km,  $\pm 50$  kV DC bipolar line. This equates to a maximum DC link voltage of 100 kV DC and from section 4.6.1 it was determined that the Wolf conductor would be the optimum conductor to use for this link.

The source impedance,  $Z_S$ , in Fig. 4.38 is equal to the p.u. impedance of the 66 kV busbar at Rundu substation, as shown in Table 4-6.  $|V_{PCCR}|$  is specified as 1.0 p.u and  $\theta_{PCCR}$  is specified as 0.1 rad.  $\theta_{PCCR}$  only determines the load current angle, as the load has a fixed power factor of 0.96. The network planner must also specify the power factor at the PCCS, the DC line resistance and the operating voltage of the DC link. The power factor is taken as unity to limit reactive power demand from Rundu substation at the sending end. The DC line resistance is equal to 2.27 p.u. and the DC link voltage equals 1.52 p.u. As previously  $S_{base} = P_{base} = 100$  MVA and  $V_{base} = 66$  kV.



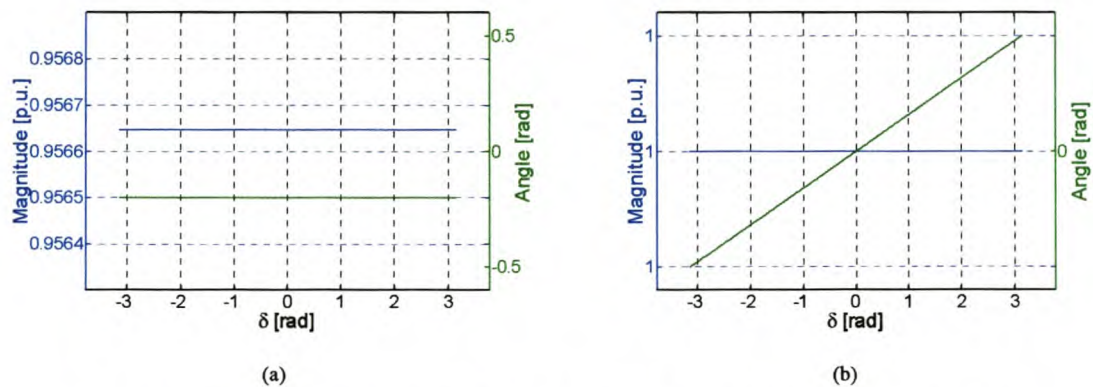
**Fig. 4.39: Voltage plots at (a) PCCS and (b) PCCR for an initial load of  $5.21 \times 10^{-3}$  p.u.**

The voltage diagrams before and after the compensator at PCCS and PCCR are shown in Fig. 4.39 for an initial load of  $5.21 \times 10^{-3}$  p.u. The voltage regulation at PCCR achieves the required 1.0 p.u., as well as the voltage regulation at the PCCS which is 0.9970 p.u.. This falls well within the NRS 048 voltage regulation specifications.



**Fig. 4.40: Power requirements of the device at (a) PCCS and (b) PCCR for an initial load of  $5.21 \times 10^{-3}$  p.u.**

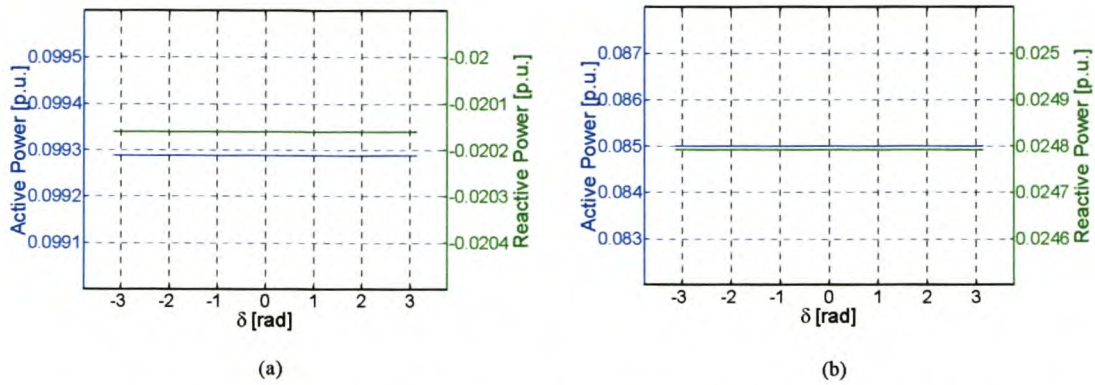
With a power factor of unity, there is no reactive power delivered by the device at PCCS. As losses are included for the DC line, the active power injected at the PCCS will not be equal to the active power received over the DC link at the PCCR, as shown in Fig. 4.40. The DC line losses can be calculated from Fig. 4.40 (a) and (b) as  $49.4 \times 10^{-6}$  p.u., which translates into 4.94 kW. For a bipolar line the loss per pole equals 2.47 kW, which corresponds with the value calculated previously for DC Wolf conductor in Table 4-15. The device at the PCCR transmits the active power received over the DC link to the load. It also delivers 0.1458 MVar reactive power to the load for a 0.96 lagging power factor.



**Fig. 4.41: Voltage plots at (a) PCCS and (b) PCCR for a final load of  $88.54 \times 10^{-3}$  p.u.**

A power factor of  $-0.98$  was chosen to achieve the desired voltage regulation at the PCCS for an anticipated final load of  $88.54 \times 10^{-3}$  p.u. The voltage regulation at the PCCR is regulated to the required 1.0 p.u., whereas the voltage regulation at the PCCS is 0.9566 p.u., which falls barely within the NRS 048 voltage regulation specification. The voltage graphs are shown in Fig. 4.41.





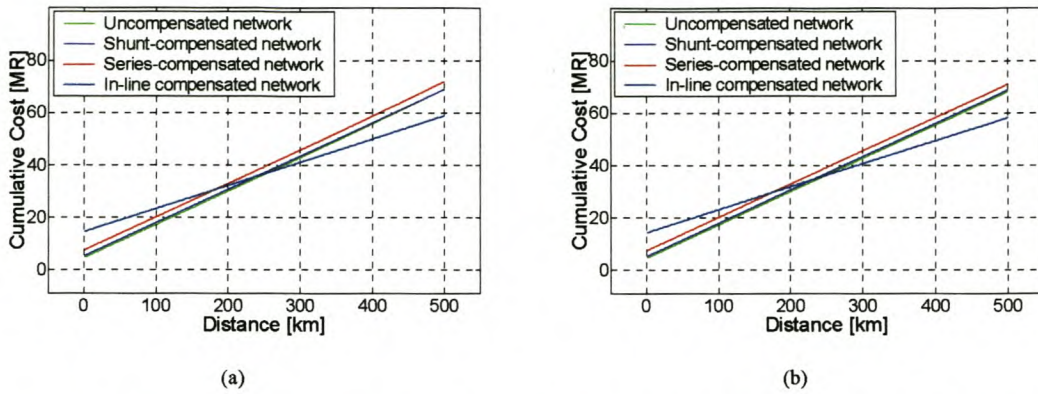
**Fig. 4.42: Power requirements of the device at (a) PCCS and (b) PCCR for a final load of  $88.54 \times 10^{-3}$  p.u.**

With a leading power factor of 0.98, there is 2.02 MVar reactive power delivered by the device at the PCCS. The losses across the DC link can be calculated from Fig. 4.42 (a) and (b) as 0.0143 p.u., which translates into 1 430 kW. The load absorbs the power delivered by the device at the PCCR.

From Fig. 4.42 the sending-end compensator rating is 0.1013 p.u., which translates into 10.13 MVA and 9.93 MW at a power factor of 0.98 leading. A change in the power factor specified at the PCCS does not alter the active compensator rating, as power demanded by the load remains constant with  $V_{PCCR}$ . The receiving-end compensator rating is 0.0885 p.u., which translates into 8.85 MVA and 8.5 MW at a power factor of 0.96 lagging.

### Determination of the break-even distance

The break-even distance between the uncompensated, shunt-compensated and series-compensated AC networks, and the DC (in-line compensated) network is shown in Fig. 4.43 (a) for final load conditions. It takes 255 km for DC transmission to break even with uncompensated AC transmission, 237 km to break even with shunt-compensated AC transmission and only 181 km to break even with series-compensated AC transmission for final load conditions.



**Fig. 4.43: Break-even distances for (a) final load conditions and (b) initial load conditions**

For initial load conditions the break-even distance is reduced even further. The break-even distance is decreased to 250 km for uncompensated AC transmission, 232 km for shunt-compensated AC transmission and 178 km for series-compensated AC transmission. These curves are shown in Fig. 4.43 (b). Beyond these distances the in-line compensated network will become more economical than the uncompensated, shunt- and series-compensated networks.

Hingorani obtained similar results for the break-even distance in the very low power transmission range (5-20 MW) [A9].



## 4.7 SUMMARY OF THE KATIMA MULILO CASE STUDY

### Summary of the line technologies

The Katima Mulilo case study served as a showcase for the practical application of the equations developed in Chapter 2 for three-phase AC and bipolar DC line technologies. A summary of the findings for the voltage regulation (VR), power losses, LCC and ATIs of different three-phase and bipolar line technologies are shown in Table 4-18.

**Table 4-18: Summary of the line technology performance in the Katima Mulilo network**

| Line Technology       |           | % VR |       | $P_{Losses}$ [kW] |       | $PV_{LCC}$ [MR] |       | ATI  |      |
|-----------------------|-----------|------|-------|-------------------|-------|-----------------|-------|------|------|
|                       |           | Min. | Max.  | Min.              | Max.  | Min.            | Max.  | Min. | Max. |
| 132 kV<br>three-phase | Pelican   | -9.8 | -15.9 | 760               | 867   | 75.50           | 76.27 | 4.3  | 4.9  |
|                       | Twin Hare | -9.3 | -15.6 | 1 307             | 1 522 | 89.50           | 90.85 | 3.6  | 4.1  |
|                       | Wolf      | -6.8 | -15.4 | 910               | 1 050 | 63.50           | 64.43 | 5.0  | 5.8  |
| ±50 kV<br>bipolar     | Rabbit    | 3.0  | 43.2  | 46                | 3 440 | 49.55           | 50.11 | 2.2  | 3.9  |
|                       | Hare      | 1.5  | 32.1  | 38                | 3 926 | 53.25           | 54.05 | 1.9  | 3.3  |
|                       | Wolf      | 1.0  | 21.3  | 35                | 2 816 | 59.15           | 59.83 | 1.3  | 2.3  |

The three-phase lines suffer from voltage regulation problems. Due to the long distance and small point load, there is a rise in the receiving-end voltage, due to the flow of line-charging current through the line inductance. The AC line technologies will, therefore, require some form of line compensation. Since inductance and capacitance have no effect on a DC line, the voltage regulation problems experienced with the AC line technologies do not arise with DC transmission. No line compensation will be required to compensate for low receiving-end voltages, because the inverter at the load end can cater for a voltage drop of more than 30% with no commutation problems (transistor-based). For the DC line technologies only the Wolf conductor will be able to transmit the full load for the lifetime of the network.

The AC line technologies will suffer maximum power losses for initial load conditions, due to large line charging currents flowing through the line inductance for

small loads over long distances. The maximum power losses for the DC line technologies are more than those for the AC line technologies, because the system voltage used for the AC technologies (132 kV phase-to-phase) is 1.52 times larger than the value that is used for the bipolar lines ( $\pm 50$  kV). Also, converter losses were included in the DC line losses.

Even though the cost of line compensation was not included in the LCC of the AC lines, the bipolar lines still prove to be a more economical solution than the more conventional three-phase AC lines. The Wolf conductor will be the cost effective solution for the AC line technologies. For the DC line technologies Wolf conductor will also be the most economic solution, because neither Rabbit nor Hare conductor will be able to transmit the full load for the lifetime of the network.

The ATIs for the line technologies are project specific with different upper and lower limits. The results are not absolute, but provide a quantitative measure of the best option for a particular line technology, i.e. AC or DC. Results from the ATI analysis for the AC and DC lines also indicate Wolf conductor to be the optimum choice in both cases.

### **Summary of the support technologies**

The Katima Mulilo network experiences voltage regulation problems for initial load conditions and the effect of shunt, series, series-shunt and in-line compensation in terms of cost and technical performance were considered for these networks. Table 4-19 is a summary of the findings for the voltage regulation of the Katima Mulilo network.



**Table 4-19: Summary of the voltage regulator performance in the Katima Mulilo network**

| Compensator  | Control         | $\delta$<br>[rad] | $ V_{PCCR} $<br>[p.u.] | Device Rating |       |        | Device Cost<br>[MR] |
|--------------|-----------------|-------------------|------------------------|---------------|-------|--------|---------------------|
|              |                 |                   |                        | [MVA]         | [MW]  | [MVar] |                     |
| None         | None            |                   | 1.17                   |               |       |        | None                |
| Shunt        | Purely reactive | 0.10              | 1.00                   | 8.60          | 0     | 8.60   | 0.70                |
| Series       | Purely reactive | -0.32             | 1.00                   | 0.197         | 0     | 0.197  | 2.9121              |
| Series-shunt | Minimum rating  | 0.70              | 1.00                   | 0.547         | 0.350 | 0.420  | None                |
| In-line      | Minimum rating  | all               | 1.00                   | 18.98         | 18.43 | 4.54   | 15.30               |

It was determined that Wolf conductor would be the optimum choice for a 500 km, 132 kV AC interconnection between Rundu substation and Katima Mulilo distribution station. The fault levels at Rundu substation are very low (high source impedance), which means that the connection of another 500 km AC line at this weak point is definitely out of the question. Compensation was, therefore, provided at Rundu substation to reduce the high source impedance, before an AC network was constructed between this point and Katima Mulilo. This compensation goes along with increased capital incurred for the AC network.

Without compensation at Katima Mulilo distribution station, the voltage rises to 1.17 p.u., which is well above the level of 1.05 required by NRS 048. All four compensator topologies solve the voltage-regulation problem. From the viewpoint of the AC network, there are two viable options, namely purely reactive shunt or series compensation.

Purely reactive shunt compensation requires 8.6 MVar reactive power absorption. The purely reactive series compensator has the lowest rating of all the options investigated. At 197 kVar it undercuts the rating of the purely reactive shunt compensator by 8.4 MVar. However, this is a very costly option, because the protection of the series device requires a very large capital outlay. Purely reactive



shunt compensation is probably the most cost-effective solution to implement for the AC network, as a 7 MVar shunt reactor bank can be installed to absorb the bulk of the reactive power. The generators at Katima Mulilo power station can then be modified as synchronous condensers to rapidly compensate the balance of the required reactive power.

For this network a series-shunt device is definitely not appropriate. The ratings of the device are shown for 0.7 rad., because a network is never operated at voltage angles larger than this angle. Although the power ratings of the device are much less than purely reactive shunt or in-line compensation, the voltage ratings are excessive. For other networks these devices can be very effective.

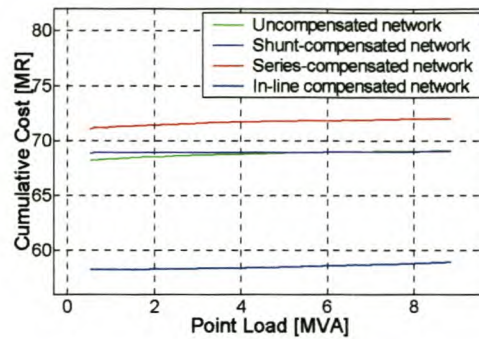
DC transmission, as an in-line technology, has the highest rating of all the options investigated. One of the distinct advantages of DC transmission (transistor-based) for this case is that it can be connected into an extremely weak AC network. No compensation was therefore required at Rundu substation to reduce the source impedance, eliminating the extra capital costs incurred with compensation at this point. It was determined that Wolf conductor would be the optimum choice of a 500 km,  $\pm 50$  kV bipolar DC line between Rundu substation and Katima Mulilo distribution station.

The capital outlay of the two converter stations required for DC transmission is the highest of all the compensator topologies. However, because the transmission distance is fairly large, the savings with the DC transmission line is enough to pay for the two converter stations. The in-line compensated network, therefore, seems to be the most effective solution in terms of cost. An economic analysis is necessary, which must include the cost of the AC and DC lines, terminals and any compensation needed for voltage support. The cost of losses must also be taken into account.

### Summary of the economic analysis

Fig. 4.44 shows the present value of LCC for the different network solutions to the Katima Mulilo case study. A summary of the economic analysis is given in Table 4-20.





**Fig. 4.44:  $PV_{LCC}$  for the different network solutions**

From Fig. 4.44 and Table 4-20 it is proved that DC transmission will be the most cost-effective solution for the Katima Mulilo case study. A series-compensated AC network will be the most expensive solution, partly due to the cost of compensation required at Rundu substation and the high capital layout of the equipment to protect the series compensator against fault conditions. Although not clearly visible in the figure, the shunt-compensated AC network will be the second most economic solution for the network lifetime.

**Table 4-20: Summary of the economic analysis for the Katima Mulilo case study**

| Network Configuration | $PV_{LCC}$ [MR] |         | $PV_{TCL}$ [MR] |
|-----------------------|-----------------|---------|-----------------|
|                       | Minimum         | Maximum |                 |
| Uncompensated         | 68.23           | 69.16   | 0.93            |
| Shunt compensated     | 68.93           | 69.07   | 0.15            |
| Series compensated    | 71.14           | 72.07   | 0.93            |
| In-line compensated   | 58.30           | 58.98   | 0.68            |

The uncompensated and series-compensated networks have the highest cost of losses, because for light loading conditions these networks experience large power losses as a result of line charging currents producing more leading reactive power than the network can absorb. For shunt-compensated networks the purely reactive compensator absorbs this leading reactive power and, therefore, this network will have the lowest cost of losses.

Beyond this simplistic economic analysis there are many details, variations and performance issues that must be considered. Detailed studies should now be conducted to determine the best solution for the Katima Mulilo case study.

## Summary of TESAT

TESAT made it possible to determine the optimum conductor for a particular AC or DC line technology within a matter of minutes. It is used as a first-assessment tool to evaluate the economic impact of different line and support technologies used in solving a network problem and to identify the most cost-effective solution. TESAT, together with PSAT, which was developed previously, will become useful tools for network planners, who need to optimise the use of line and support technologies to limit capital outlay and stretch networks up to thermal capacity. The innovative three-dimensional graphs provide additional insight into the voltage regulation, power losses, LCC and ATIs of different line technologies for various point load sizes and distances. Two-dimensional cuts enhance the understanding of the line technology performance even further.



## **CHAPTER 5 OTHER WORK AND FUTURE DEVELOPMENTS**

### **5.1 PSAT – RETICMASTER INTERFACE**

#### **5.1.1 Implementation**

### **5.2 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE**

### **5.3 FUTURE DEVELOPMENTS**

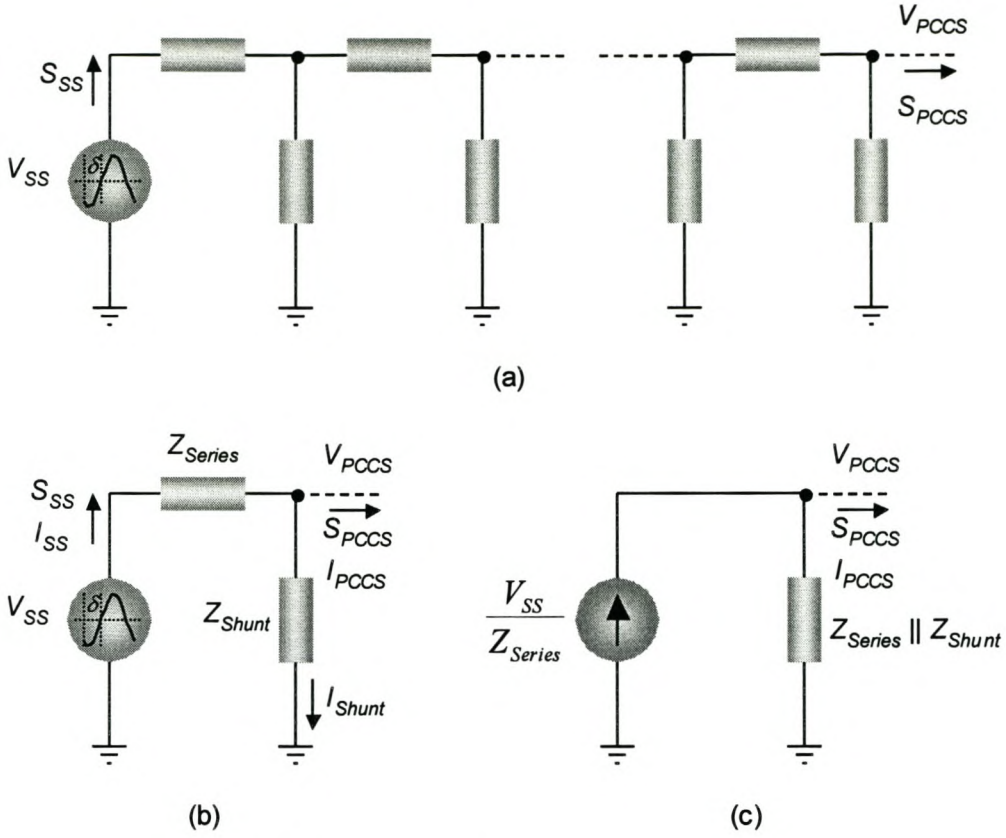
## 5.1 PSAT – RETICMASTER INTERFACE

Provision is made in PSAT for two sources of network parameters. For a sub-transmission network the network parameters are calculated directly from the line parameters. Sub-transmission networks connect substations with substations or loads. The equivalent network is constructed from the known sending- or receiving-end voltage and the calculated line parameters. The calculation of the OLP as well as the interface with PSAT was discussed in detail in section 2.2.

Many loads are connected along the lines in reticulation networks. It was proven previously [C1] that the Thévenin equivalent network can be calculated for such a network, but is very tedious. The alternative is to obtain the electrical characteristics of the network from a ReticMaster model and use these to construct the sending- and receiving-end circuits. ReticMaster was developed by Inspired Interfaces to quickly and efficiently model radial networks from high-voltage source distribution down to individual low-voltage consumers [B18].

Fig. 5.1 portrays the sending end of the network, with many loads connected to the reticulation line.  $V_{SS}$ ,  $S_{SS}$ ,  $V_{PCCS}$ , and  $S_{PCCS}$  can be obtained from network simulation data at any two nodes in ReticMaster. With these parameters known, the lines and loads connected between these nodes can be represented by a single series and shunt impedance, as shown in (b). This method, however, is restricted to constant impedance loads between the nodes concerned. Since most of the loads in rural areas are constant impedances, this restriction is acceptable.





**Fig. 5.1: Modelling of a reticulation network**

From (b),  $Z_{Series}$ , and  $Z_{Shunt}$  are calculated as follows:

$$\begin{aligned} V_{Z_{Series}} &= V_{SS} - V_{PCCS} \\ I_{SS} &= \frac{S_{SS}^*}{V_{SS}^*} \end{aligned} \quad (5-1)$$

$$Z_{Series} = \frac{V_{Z_{Series}}}{I_{SS}} = \frac{|V_{SS}|^2 - V_{PCCS} V_{SS}^*}{S_{SS}^*}$$

$$\begin{aligned} I_{PCCS} &= \frac{S_{PCCS}^*}{V_{PCCS}^*} \\ I_{Shunt} &= I_{SS} - I_{PCCS} \end{aligned} \quad (5-2)$$

$$Z_{Shunt} = \frac{V_{PCCS}}{I_{Shunt}} = \frac{|V_{PCCS}|^2 V_{SS}^*}{S_{SS}^* V_{PCCS}^* - S_{PCCS}^* V_{SS}^*}$$

With  $V_{SS}$ ,  $Z_{Series}$  and  $Z_{Shunt}$  known, a Thévenin-Norton source transformation results in (c). A Norton-Thévenin source transformation finally takes (c) to the sending end of PSAT, where  $V_S$  and  $Z_S$  are given by equations (5-3) and (5-4):

$$V_s = \frac{V_{SS}}{Z_{Series}} \left( \frac{Z_{Series} Z_{Shunt}}{Z_{Series} + Z_{Shunt}} \right) \quad (5-3)$$

$$= \frac{V_{SS} Z_{Shunt}}{Z_{Series} + Z_{Shunt}}$$

$$Z_s = Z_{Series} \parallel Z_{Shunt} \quad (5-4)$$

$$= \frac{Z_{Series} Z_{Shunt}}{Z_{Series} + Z_{Shunt}}$$

This method is not only restricted to the sending end of the network, but can also be applied to the receiving end.

### 5.1.1 Implementation

In this section the interface between PSAT and ReticMaster is discussed. Firstly a user report is created in ReticMaster, which is then exported as a text file. The electrical characteristics of the network are then extracted from this file by the PSAT-ReticMaster interface.

The user can define a report using the available fields by selecting *Tools/Options* from the main menu in ReticMaster and clicking on the *User Report Page* tab shown in Fig. 5.2. Multiple fields can be selected and moved to the *Report Fields* listbox. The order of how the fields appear in the report can also be changed using the move buttons [B18]. The fields that the PSAT-ReticMaster interface requires to calculate  $V$  and  $Z$  are displayed in the *Report Fields* listbox in Fig. 5.2. A file, *Report.rpt*, was created to ensure that the necessary fields would appear in the correct order in the *Report Fields* listbox. Selecting the *Retrieve a user report from a disk* button can retrieve this report.



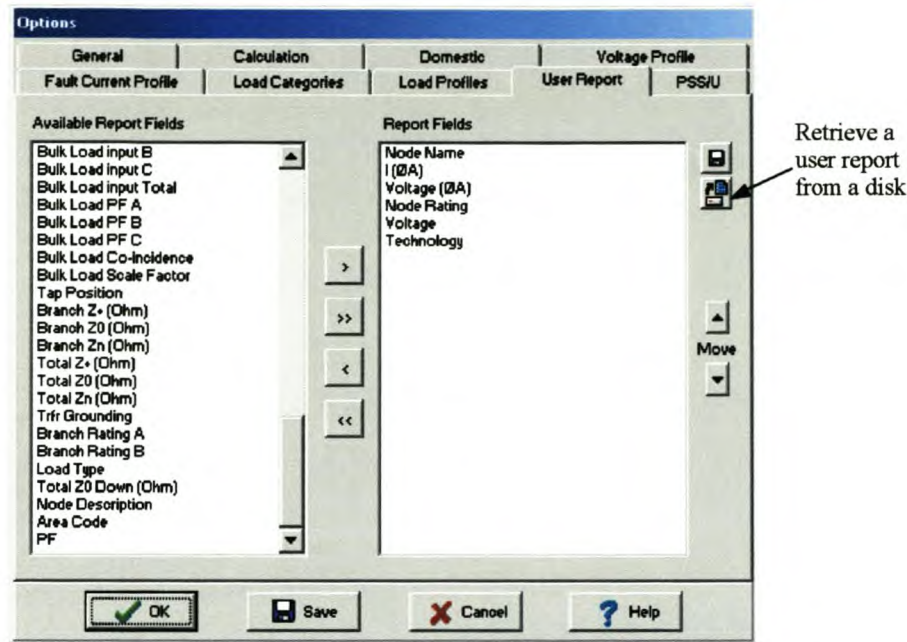


Fig. 5.2: User Report Page tab in ReticMaster

The user must specify the fields for the report before selecting *Tools/Report* from the main menu. A report can then be generated for either a portion of, or the complete, network. The user has the option of clicking on a required field and the report will be sorted according to this field. Buttons are available to select either ascending or descending order. Selecting *Saves the current file* button will export the user report to a text file for further analysis in the PSAT-ReticMaster interface.

Saves the current file

| Node Name | I (0A) (A)      | Voltage (0A)    | Node Rating            | Voltage | Technology   |
|-----------|-----------------|-----------------|------------------------|---------|--------------|
| WINTERO   | 103.02 <-14.83° | 102.50 <-0.00°  | Source - WINTERO       | -       | Star System  |
| T191_13   | 1.05 <-14.16°   | 100.32 <149.36° | 11kV                   | 11kV    | Delta System |
| T191_20   | 2.22 <-14.59°   | 100.31 <149.36° | 11kV                   | 11kV    | Delta System |
| T305_10   | 0.32 <-13.72°   | 102.76 <147.75° | 11kV                   | 11kV    | Delta System |
| T69_17    | 1.59 <-38.31°   | 98.90 <148.56°  | 11kV                   | 11kV    | Delta System |
| T69_12    | 0.59 <-17.70°   | 98.90 <148.56°  | 11kV                   | 11kV    | Delta System |
| T69_19    | 0.59 <-17.70°   | 98.90 <148.56°  | 11kV                   | 11kV    | Delta System |
| T965_3    | 0.20 <-14.26°   | 99.01 <147.20°  | 11kV                   | 11kV    | Delta System |
| T414_1    | 0.03 <-79.03°   | 102.75 <-0.76°  | 1000kVA 22kV/11kV Star | 11kV    | Star System  |
| T268_12   | 0.32 <-18.64°   | 101.16 <149.56° | 11kV                   | 11kV    | Delta System |
| T191_23   | 0.73 <-12.11°   | 100.32 <149.36° | 11kV                   | 11kV    | Delta System |
| RM473     | 20.58 <-15.87°  | 100.04 <-3.11°  | 2000kVA 11kV/11kV Star | 11kV    | Star System  |

Fig. 5.3: The user report in ReticMaster

The interface to extract the electrical characteristics of a reticulation network from the ReticMaster user report is shown in Fig. 5.4. The user enters the ReticMaster report name, which is retrieved when the *Load* button is selected. The interface calculate  $V$

and  $Z$  between the source node of the reticulation network and a user-selected load node if the *ReticMaster Source Node* radio button is selected. Different load nodes are selected by clicking on the node name in the *ReticMaster Load Node* listbox. If the *Alternative Source Node* radio button is selected, the interface will calculate  $V$  and  $Z$  between any two load or source nodes.  $|V|$  and  $Z$  can be sent to either the sending or the receiving end in the PSAT input interface.

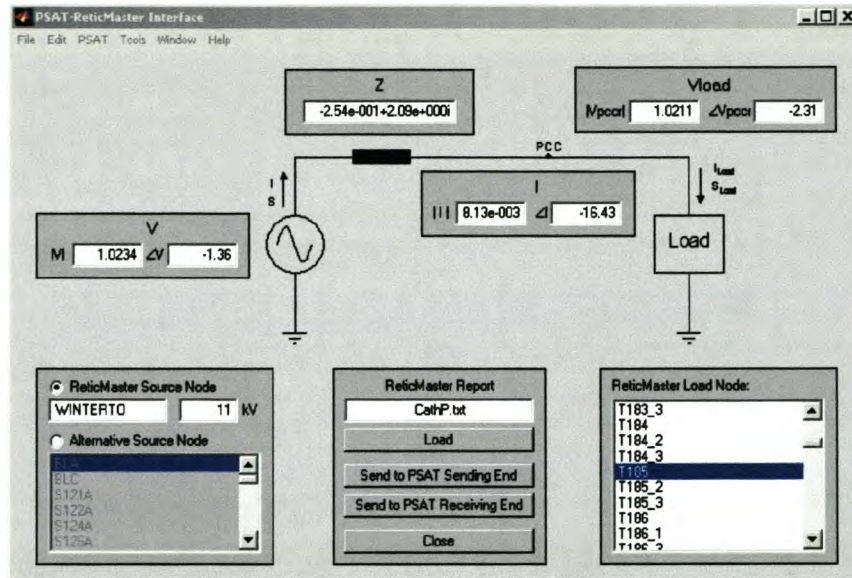


Fig. 5.4: Interface to extract network parameters from ReticMaster

## Software development

The PSAT-ReticMaster interface was, just like the OLP interface of PSAT, implemented in Matlab. The reasons for choosing Matlab and the use of Matlab GUIs were discussed in section 2.2.4.

The equations to calculate  $V$  and  $Z$  are implemented in *interface.m*, of which the full program listing is available in Addendum B2.

ReticMaster version 4.2 was used to create the user report.



## 5.2 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE

A database infrastructure has been developed to link the ratings and configurations determined by PSAT to real devices [C2]. Much data have been sourced and are currently fed into this library. The user interface of the database is shown in Fig. 5.5.

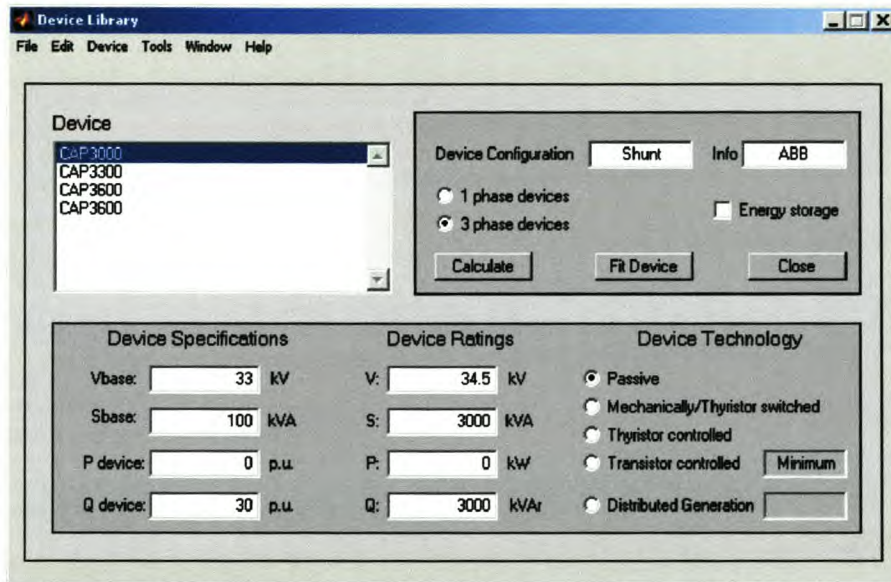


Fig. 5.5: Device library window

The following parameters are sent from PSAT to the device library:

- The device configuration – series, shunt, series-shunt or in-line;
- The value of  $P_{device}$  and  $Q_{device}$  in p.u. The device rating is determined by the specific network configuration and parameters, as well as the device configuration and control method;
- If the device requires energy storage, PSAT will mark the *Energy storage* checkbox. An energy-storage device can be connected to the DC bus of any transistor-controlled device [C19]. A device with energy storage or a distributed generator can supply both active and reactive power;
- For single-phase and three-phase networks PSAT enables the *1 phase devices* and *3 phase devices* radio buttons, respectively.

The user must enter the base values for  $V_{base}$  and  $S_{base}$ . PSAT determined the device configuration, but the user must decide the device technology. The network device technology will determine the reaction time and variation of the compensating power.



Depending on dynamic needs, the user must decide among the following technologies:

- Passive technology, such as shunt capacitors and reactors;
- Mechanically or thyristor switched technology, such as switched shunt capacitors and reactors or a tap-changing transformer;
- Thyristor-controlled technology, such as a Static Var Compensator (SVC), a TCSC or a HVDC line;
- Transistor-controlled technology, such as Static Synchronous Compensator (STATCOM) or a Dynamic Voltage Restorer (DVR).

Distributed generation can be evaluated as a network support technology and is a special case of shunt compensation. It is important not to confuse this technology with a distributed generator as a power source at the receiving end of a network.

Selecting the *Calculate* button will automatically calculate the actual device ratings for a specified  $S_{base}$  and  $V_{base}$ . The values are displayed in the *Device Ratings* column of the interface. Pressing the *Fit Device* button will search the database for all similar devices with equal or larger ratings. The names of all the devices found will be displayed in the *Device* listbox. The ratings of these devices can be displayed in the *Device Ratings* column by clicking on the device names in the listbox. The *Info* box will display any additional data about the selected device, e.g. a manufacturer's name such as ABB. If no devices could be found for the specified device configuration, rating and technology, the message "no device" will be displayed in the *Info* box.

The user can add new devices to the database by selecting *Device/Add Device* from the main menu in the device library interface. Selecting *Device/Modify Device* from the main menu can modify existing devices. The add device and modify device interfaces of the device library are shown in Fig. 5.6.

## Implementation

The device library interface was also implemented in Matlab. A discussion on the reasons for choosing Matlab and the use of Matlab GUIs was included in section 2.2.4.



(a)

(b)

**Fig. 5.6: a) Add device and b) modify device interfaces**

The program for the device library interface, as well as the add device and modify device interfaces, is implemented in *device.m*. The program starts by importing the device parameters send over by PSAT. It also imports the user-defined base values and device technologies. Next, all the devices in the database for the defined device technology and configuration are compared. All the devices found with equal or larger ratings are extracted from the database and listed in the device interface. The full program listing for *device.m* is included in Addendum B3.

## 5.3 FUTURE DEVELOPMENTS

### Future developments in PSAT interfaces

The OLP interface of PSAT is a complete package for calculating the series impedance and shunt admittance for balanced three-phase, three-wire lines, where it is assumed the lines are perfectly transposed. The following future extensions are proposed for the OLP interface of PSAT:

- Extension of the conductor database to include other conductor types as well. These other conductor types include AAC, AAAC, aluminium conductor alloy-reinforced (ACAR), and aluminium-alloy conductor steel-reinforced (AACSR). Aerial bundled conductor (ABC) and steel conductor also need to be included in the database;
- The calculation of AC and DC resistance values at different operating temperatures for the various conductor types, as was discussed in detail in section 2.2.1;
- Equations were developed in sections 2.2.1 and 2.2.2 to calculate the phase and sequence impedances and admittances for a three-phase line with neutral conductors. This included the calculation of the series phase impedance matrix  $Z_p$  (2-31) and the shunt phase admittance matrix  $Y_p$  (2-54) for untransposed three-phase lines with neutral conductors. Equations were also developed to calculate the phase and sequence impedances and admittances of completely transposed lines. These equations can be implemented in the OLP interface to calculate the phase and sequence impedances of untransposed as well as transposed three-phase lines with and without neutral conductors. The phase and sequence values can also be calculated for other line technologies, such as phase-phase, single-phase or SWER, using the matrices developed in section 2.2. Because the effect of neutral wires is included, both balanced and unbalanced conditions can be considered.

These extensions will enable the OLP interface to accurately calculate the line parameters for any line technology with various conductor types, bundle selections and tower configurations.



## **Future developments in TESAT interfaces**

It was shown extensively that TESAT could be used to do voltage regulation analysis and the calculation of line losses and transmission costs. Presently only three-phase lines are analysed in the AC interface of TESAT and bipolar lines in the DC interface. A technology index to compare different line technologies on a per project basis was also implemented in these interfaces. The objective of these technology indexes is to select the optimum conductor for a specific AC or DC line technology, based on the lowest transmission cost, best emergency performance, stability of transmission and quality of supply. The following future developments are proposed for the AC and DC interfaces of TESAT:

- Implementation of equations to calculate the voltage regulation, line losses and transmission costs of other line technologies, such as phase-to-phase, single-phase and SWER lines in the AC interface and monopolar lines in the DC interface of TESAT;
- The inclusion of these other line technologies in the AC and DC technology indexes, including the development of equations to calculate the emergency and stability performance of these line technologies;
- The development of equations to analyse the effect of network support devices on the voltage regulation, power losses and transmission costs of AC line technologies;
- The extension of the AC technology index to include the effect of line compensation. The analysis equations were documented in great detail in sections 2.6.1 and 2.7.1.

These extensions to the TESAT interfaces will enable network planners to determine the optimum line technology and conductor when designing new networks or upgrading existing networks. It will also allow planners to consider network support devices in order to operate lines closer to their thermal limits.

## **SWER as an alternative for low power, long distance transmission**

The main benefit of SWER lines is that a single conductor is used, which means savings on line hardware. A SWER line can, therefore, be constructed at the fraction of the cost of conventional three-phase overhead lines. This section discusses the applicability of SWER networks for the Katima Mulilo case study. The phase and

sequence impedances are calculated and the voltage regulation performance of the lines are analysed in TESAT.

The following two conductors are evaluated with a 132 kV Kamerad tower structure shown in Fig. 4.2:

- Wolf conductor
- Pelican conductor.

The AC resistance values at 40°C were calculated from the DC resistance values at 20°C and are shown in Table 4-1.

From section 2.2, the positive-sequence impedance for the SWER lines can be calculated as follows:

$$\begin{aligned}
 \hat{Z}_1 &= \hat{Z}_S - \hat{Z}_M \\
 &= Z_{aaeq} \\
 &= r_{aa} + r_{a'} + j\omega 2 \times 10^{-4} \ln \frac{D_{aa'}}{D_{aa}} \\
 &= 0.2585 + j0.7370 \quad [\Omega/\text{km}] \quad (\text{Wolf}) \\
 &= 0.1784 + j0.7323 \quad [\Omega/\text{km}] \quad (\text{Pelican})
 \end{aligned} \tag{5-5}$$

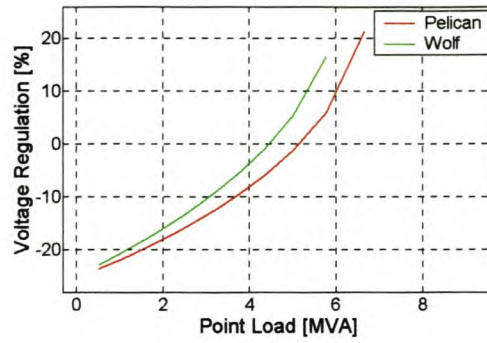
and the shunt admittance as:

$$\begin{aligned}
 \hat{Y}_1 &= \hat{Y}_S - \hat{Y}_M \\
 &= j\omega C_{aa} \\
 &= j\omega \frac{2\pi\epsilon}{\ln(D_{eq}/r)} \\
 &= 2.25 \quad [\mu\text{S}/\text{km}] \quad (\text{Wolf}) \\
 &= 2.29 \quad [\mu\text{S}/\text{km}] \quad (\text{Pelican})
 \end{aligned} \tag{5-6}$$

The values of series impedance and shunt capacitance for SWER lines will be larger than that for three-phase lines for the same choice of conductors. The reason for this is that the equivalent area perpendicular to the flux linkage is much larger.

The voltage regulation performance of the lines were analysed in TESAT for the Katima Mulilo network and the RXB values calculated in equations (5-5) and (5-6). A base value of 33⅓ MVA (100 MVA/3) is used for  $S_{base}$  and 76 kV (132 kV/ $\sqrt{3}$ ) for  $V_{base}$ . The results of the analysis are shown in Fig. 5.7.





**Fig. 5.7: Voltage regulation of 76.21 kV SWER lines as a function of load growth**

The Wolf line will be able to conduct the full load until the 17<sup>th</sup> year and the Pelican line until the 18<sup>th</sup> year. Neither line will be able to conduct the full load for the lifetime of the network. Fig. 5.7 and Table 5-1 shows that for initial loading conditions high line-end voltages are experienced due to the Ferranti effect. Reactors will therefore be required for voltage control at the receiving end.

**Table 5-1: Summary of the voltage regulation performance**

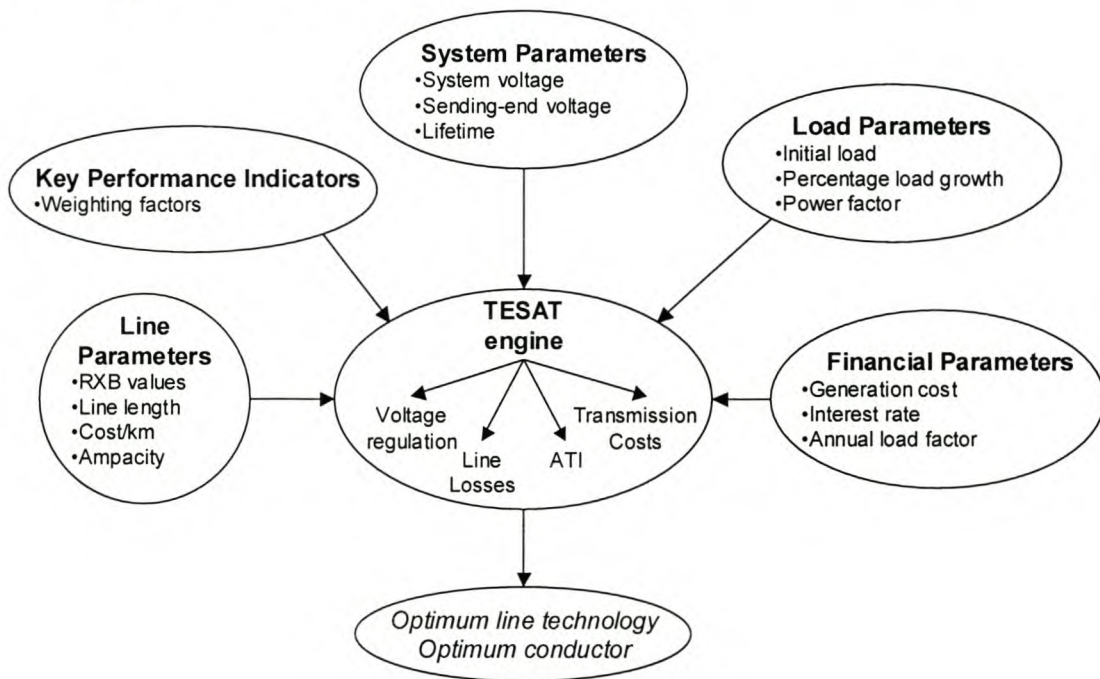
| 76 kV<br>AC SWER | % Voltage Regulation |         |
|------------------|----------------------|---------|
|                  | Minimum              | Maximum |
| Pelican          | -23.57               | 21.37   |
| Wolf             | -22.77               | 16.55   |

The SWER networks can be upgraded to three-phase as the loading increases, allowing a phase implementation of infrastructure and delayed capital investment. For the Katima Mulilo case study it is, therefore, very advantageous to design the network to be extremely “light” initially with high flexibility by means of practical and cost effective upgrade paths to ensure that optimum use is made of resources and capital.

## **CHAPTER 6 CONCLUSION**



In today's economic environment it is important to find the most effective solution in terms of cost and technical performance when planning new networks or upgrading existing networks. The application of line and support technologies in the design process can help achieve this goal. Network planners need tools to analyse and implement such line and support technologies. They also need to be trained in such analysis and implementation. An analysis tool, PSAT, has been developed previously to model various network configurations, network problems and support technologies as an aid to power system engineers for implementing network technologies [C1]. One of the objectives of this thesis was to document the development of a software tool, TESAT, to assist the network planner in selecting the optimum conductor and line technology for a network. If network support is required, i.e. to solve voltage regulation problems in an attempt to stretch the network up to thermal capacity, TESAT can be used to evaluate the economic impact of the line and support technology on network costs.



**Fig. 6.1: Inputs and outputs of the TESAT engine**

A diagram of the technical and financial inputs and outputs of the TESAT engine is shown in Fig. 6.1. The equations of the TESAT engine must consider the following inputs:

- Line parameters
- System parameters
- Load parameters

- Financial parameters
- Key performance indicators (KPIs).

The determination of the line parameters for three-phase AC and bipolar DC lines was discussed in Chapter 2. The equations for voltage regulation, transmission losses and LCC for these line technologies with different conductors were subsequently discussed. The cost of support technologies, if any, was included in the transmission costs of the line technologies.

Having determined the voltage regulation, losses and life cycle costs, the ATI was calculated to compare the various line technologies on a per project basis. For AC line technologies the KPIs to consider were voltage regulation, capacity, stability and voltage regulation. The DC ATI included the LCC, capacity and voltage regulation KPIs. The KPIs were summated using weighting factors to evaluate the overall technical and economical performance of a line technology.

The thermal load reach and transfer capabilities of different three-phase AC and bipolar DC lines at distribution voltage levels were determined in Chapter 3. The formula to calculate the load reach and transfer capabilities of these line technologies were derived from the voltage regulation equations developed in Chapter 2.

The Katima Mulilo case study was presented in Chapter 4 to show the practical application of TESAT. The case study showed that TESAT uses basic information to quickly calculate the voltage regulation, losses and LCC of different conductors from an AC and DC perspective in order to select the optimum conductor and line technology for the task at hand. The combination of network support technologies and line technologies to find the most effective solution in terms of cost and technical performance was also addressed by the case study. Ultimately, the case study showed that for low-power transmission over long distances, DC transmission at lower voltage levels is more economical than AC transmission at higher voltage levels. Apart from being technically more reliable than the AC interconnection, DC transmission will cost about MR 10 (15%) less than the most viable AC alternative for this particular case study.



Chapter 5 discussed the interfaces between PSAT and the real world. The interface between PSAT and ReticMaster was presented, as well as the development of the support technology and dispersed generation database. Future developments in the PSAT interfaces and TESAT were subsequently discussed.

TESAT provides a sturdy platform on which future developments such as the technical and economic performance of alternative line technologies can be modelled. However, TESAT can already operate as an analysis aid to determine the optimum conductor and line technology (three-phase or bipolar) for a network, and the economic impact of combining line and support technologies on network costs.

## REFERENCES



## Journals and conference papers

- [A1] F.Engelbrecht, F.J. Rossouw and H.J. Beukes, *The Development of a Power System Analysis Tool*, 2001 SAUPEC, pp. 119-124, January 2001.
- [A2] F.J. Rossouw and H.J. Beukes, *Analysis of Voltage Regulation and Network Support Technologies*, IASTED International Conference, pp. 594-602, September 2000.
- [A3] V.T. Morgan, *Electrical Characteristics of Steel-cored Aluminium Conductors*, IEE Proceedings, Vol. 112, No. 2, pp. 325-334, February 1965.
- [A4] C.F. Price and R.R. Gibbon, *Statistical Approach to Thermal Rating of Overhead Lines for Power Transmission and Distribution*, IEE Proceedings, Vol. 130, September 1983.
- [A5] J. R. Carson, *Wave Propagation in Overhead Wires with Ground Return*, Bell System Technical Journal, Vol. 5., pp. 539-554, 1926.
- [A6] C.W. Brice, *Voltage-Drop Calculations and Power-Flow Studies for Rural Electric Distribution Lines*, IEEE Transactions on Industry Applications, Vol. 28, No. 4, July/August 1992.
- [A7] D. Povh, *Use of HVDC and Facts*, Proceedings of the IEEE, Vol. 88, No. 2, pp. 235-245, February 2000.
- [A8] N.G. Hingorani, *High Voltage DC Transmission: A power electronics workhorse*, IEEE Spectrum, pp. 63-72, April 1996.
- [A9] N.G. Hingorani, *DC Technology for Rural Transmission*, CIGRE international colloquium on HVDC and FACTS, paper 3.1, Johannesburg 1997.
- [A10] B. Meyer and H.J. Beukes, *Electronic Voltage Regulation of MV and LV Power Distribution Networks*, Energize journal, pp. 24-26, November/December 2000.
- [A11] Working Group 12 Cigre, *Probabilistic Determination of Conductor Current Ratings*, Electra Number 164, pp. 103 - 119, February 1996.
- [A12] F. Engelbrecht and H.J. Beukes, *Analysis of AC and DC Transmission at Distribution Voltage Levels*, IASTED PES 2001, Rhodes, Greece, July 2001.
- [A13] L.L. Dvorak, D.C. Dawson, R.W. Dempsey, H. Disante, C.M. Gadsden, J.D. III Huddleston and L.J. Schulze, *Summary of the 'Guide for the Protection of Shunt Reactors' ANSI C37.109*, IEEE Transactions on Power Delivery, pp. 116-118, January 1991.

- [A14] G. Asplund, K. Eriksson, K. Svensson, *DC transmission based on Voltage Source Converters*, CIGRE SC14 Colloquium, Johannesburg, South Africa, 1997.
- [A15] G. Asplund, K. Eriksson and B. Drugge, *Electric power transmission to distant loads by HVDC Light*, Distribution 2000, Sydney, Australia, 1997.
- [A16] U. Axelsson, A. Holm, C. Liljegren, K. Eriksson and L. Weimers, *Gotland HVDC Light Transmission – World's First Commercial Small Scale DC Transmission*, CIRED Conference, Nice, France, May 1999.
- [A17] R. Grünbaum, B. Halvarsson and A. Wilk-Wilczynski, *FACTS and HVDC Light for Power System Interconnections*, Power Delivery Conference, Madrid, Spain, September 1999.
- [A18] A.E. Hammad and W.F. Long, *Performance and Economic Comparisons between Point-to-Point HVDC Transmission and Hybrid Back-to-Back HVDC/AC Transmission*, IEEE/PES 1989 Transmission and Distribution Conference, New Orleans, 1989.

### **Books and manuals**

- [B1] C. Adamson and N.G. Hingorani, *High Voltage Direct Current Power Transmission*, Garraway Ltd, London, 1960.
- [B2] J. Arrillaga, *High Voltage Direct Current Transmission 2<sup>nd</sup> Edition*, IEE Power and Energy Series 29, England, 1998.
- [B3] N.G. Hingorani and L. Gyugi, *Understanding FACTS*, IEEE Press, New York, 2000.
- [B4] A.E. Guile, W. Paterson, *Electrical Power Systems (Second Edition)*, Pergamon Press, Oxford, 1977.
- [B5] J.D. Glover and M. Sarma, *Power System Analysis and Design (Second Edition)*, PWS Publishing Company, Boston, 1994.
- [B6] Westinghouse Electric Company, *Electrical Transmission and Distribution Handbook 4<sup>th</sup> Edition*, R.R. Donnelley and Sons Company, Chicago, 1964.
- [B7] J.J. Grainger and W.D. Stevenson, *Power System Analysis*, McGraw-Hill Inc, 1994.
- [B8] C.F. Wagner and R.D. Evans, *Symmetrical Components*, McGraw-Hill Inc., New York, 1933.



- [B9] P.M. Anderson, *Analysis of Faulted Power Systems*, Iowa State Press, Ames, 1973.
- [B10] Turan Gönen; *Modern Power System Analysis*, John Wiley and Sons Inc, New York, 1988.
- [B11] P. Kundur; *Power System Stability and Control*, McGraw-Hill Inc., New York, 1994.
- [B12] *Building GUIs with Matlab*, The Mathworks Inc., Natick M.A, 1997.
- [B13] X. Wang and J.R. McDonald, *Modern Power System Planning*, McGraw-Hill Inc., Cambridge, 1994.
- [B14] H. Lee Willis, *Power Distribution Planning Reference Book*, Marcel Dekker, Inc., New York, 1997.
- [B15] Regulation 9 of Electricity Act, 1987 (as amended in January 1996).
- [B16] NRS 048, Electricity Supply – Quality of Supply, Southern African Bureau of Standards, Pretoria, 1996.
- [B17] T.J.E. Miller, *Reactive Power Control in Electric Systems*, John Wiley & Sons, New York, 1982.
- [B18] *ReticMaster Quick Reference Guide*, Inspired Interfaces Ltd., Durban, South Africa, 2000.

### Other references

- [C1] F.J. Rossouw, *Analysis of Voltage Regulation and Network Support Technologies*, Masters thesis, University of Stellenbosch, October 2000.
- [C2] H.J. Beukes, B. Meyer and F.J. Rossouw, *Network Options and Integration: Volume 3*, Eskom SAPSSI Research Report, Stellenbosch, October 2000.
- [C3] C.F. Smit, *Calculation of Resistance of Overhead Line Conductors*, March 1996.
- [C4] Aberdare Cables Africa Ltd, *Overhead Aluminium Conductor Tables*.
- [C5] V.T. Morgan, *The Thermal Rating of Overhead Line Conductors, Part 1: The Steady State Thermal Model*, Electric Power Systems Research, pp. 119-139, 1982.
- [C6] B. Meyer, *Opportunities for In-line, Transistor Based Technologies on MV and LV Power Distribution Networks*, Masters thesis, University of Stellenbosch, October 2000.
- [C7] C.F. Smit, *Transmission Line Loadflow*, Draft document, February 2001.

- [C8] M.C. Mpye, *Energy Loss Management Directive*, Eskom Energy Management Group, April 1999.
- [C9] J. Jordaan, *Conductor Optimisation*, Eskom Transmission Line Technology Department, June 1996.
- [C10] G. Stanford, *Distribution Standard: Part 6: Sub-transmission lines, Section 2: Conductors*, Eskom Distribution Technology Group, September 1999.
- [C11] EED 15/6/1-1:1970, *Thermal Limits of Transmission Line and Busbar Conductors*, 1970.
- [C12] C.F. Smit, *Conductor.xls*, Excel Spreadsheet.
- [C13] G. Stanford, *Eskom Standard, Determination of Conductor Current Ratings in Eskom*, Eskom Distribution Technology Group, June 2000.
- [C14] M. Manchen, F. Engelbrecht, Private discussions, Nampower Head Quarters in Windhoek, Namibia, February 2001.
- [C15] G. Stanford, *Distribution Specification, Specification for Phase Conductor for Distribution Lines*, Eskom Distribution Technology Group, February 2001.
- [C16] I.A. Ferguson, *Distribution Standard, Part 1: Planning guidelines, Section 42: Tool – Electrification technology selection*, Eskom Distribution Technology Group, February 2000.
- [C17] EMCON Consulting Engineers, *Rural Electricity Distribution Master Plan for Namibia, Volume 2: Regional Planning Report for the Caprivi Region*, Internal Nampower Report, Namibia, August 2000.
- [C18] Africa Grid Planning, *Power System Data Base*, Internal Nampower Report, Namibia, June 1999.
- [C19] J.H.R. Enslin, H. du T. Mouton, D.D. Bester, A.D. le Roux and A.J. Visser, *Control and Implementation Considerations of a 2 MVA Power Quality Compensator*, Eskom SAPSSI Research Report, Stellenbosch, October 1999.



# **ADDENDUM A CONDUCTOR DATA AND CONVERTER STATION COSTS**

## **A.1 CONDUCTOR DATA**

## **A.2 ESTIMATED CONVERTER STATION COSTS**

A.1 CONDUCTOR DATA

Table A-1: Conductor Data [C4]

| Conductor<br>Name | Material<br>Type | Stranding and Wire<br>Diameter [mm] |         | Overall<br>Diameter<br>[mm] | Max DC<br>Resistance<br>[Ω/km, 20 °C] | DC<br>Rating<br>[A] |
|-------------------|------------------|-------------------------------------|---------|-----------------------------|---------------------------------------|---------------------|
|                   |                  | Aluminium                           | Steel   |                             |                                       |                     |
| Ash               | AAAC             | 19/3.48                             | —       | 17.40                       | 0.1840                                | 467                 |
| Chickadee         | ACSR             | 18/3.77                             | 1/3.77  | 18.87                       | 0.1427                                | 530                 |
| Fox               | ACSR             | 6/2.79                              | 1/2.79  | 8.37                        | 0.7822                                | 190                 |
| Gopher            | ACSR             | 6/2.36                              | 1/2.36  | 7.08                        | 1.0933                                | 150                 |
| Hare              | ACSR             | 6/4.72                              | 1/4.72  | 14.16                       | 0.2733                                | 360                 |
| Kingbird          | ACSR             | 18/4.78                             | 1/4.78  | 23.88                       | 0.0894                                |                     |
| Magpie            | ACSR             | 3/2.118                             | 4/2.188 | 6.35                        | 2.7070                                | 92                  |
| Mink              | ACSR             | 6/3.66                              | 1/3.66  | 10.98                       | 0.4546                                | 260                 |
| Oak               | AAAC             | 7/4.65                              | —       | 13.95                       | 0.2790                                | 359                 |
| Pelican           | ACSR             | 18/4.14                             | 1/4.14  | 20.70                       | 0.1189                                | 600                 |
| Rabbit            | ACSR             | 6/3.35                              | 1/3.35  | 10.05                       | 0.5426                                | 240                 |
| Squirrel          | ACSR             | 6/2.11                              | 1/2.11  | 6.33                        | 1.3677                                | 130                 |
| Tern              | ACSR             | 45/3.38                             | 7/2.25  | 27.00                       | 0.0718                                | 830                 |
| Wolf              | ACSR             | 30/2.59                             | 7/2.59  | 18.13                       | 0.1828                                | 470                 |



## A.2 ESTIMATED CONVERTER STATION COSTS

Each converter station consists of a voltage source converter, a DC bus, an AC filter and the converter control. The estimated converter station costs are shown in Table A-2.

**Table A-2: Estimated converter station costs**

| Description  | Quantity | Price/Unit [R] | Total [R]     |
|--|----------|----------------|---------------|
| Switches   |          |                |               |
| 3.3 kV 400 A Dual Pack IGBTs                         | 182      | 10 200.00      | 1 856 400.00  |
| Heat Sinks   |          |                | 464 100.00    |
| DC Bus   |          |                |               |
| DC Capacitors (3 300 $\mu$ F, 450 V <sub>DC</sub> )  | 2 500    | 400.00         | 1 000 000.00  |
| Filter Components                                    |          |                |               |
| Filter Capacitors (50 $\mu$ F, 440 V <sub>AC</sub> ) | 16 434   | 130.00         | 2 136 420.00  |
| Filter Inductors                                     | 6        | 100 000.00     | 600 000.00    |
| Control and Measurement Equipment                    |          | 2 000 000.00   | 2 000 000.00  |
| Auxiliary Equipment                                  |          | 2 000 000.00   | 2 000 000.00  |
| Sub-total 1  |          |                | 10 556 920.00 |
| Other Costs  |          |                | 5 028 460.00  |
| Total Cost   |          |                | 15 085 380.00 |
| Total Cost/MW  |          |                | 1 774 751.00  |

The heart of each converter station is a three-phase voltage source converter with series connected IGBTs in each valve. For a maximum DC link voltage of 100 kV, each converter valve will consist of a minimum of 31 series connected 3.3 kV, 400 A dual pack IGBT switches. Every IGBT is provided with an antiparallel diode. Turn on/off of each IGBT is ordered via an optical link from the control equipment at ground potential. The semiconductors are cooled with heat sinks. The cost of these heat sinks was estimated as 25% of the total cost of the IGBT switches.

The primary objective of the DC capacitor bank is to provide a low inductive path for the turn-off current, and energy storage to be able to control the power flow [A14]. The DC capacitor bank consists of 10 parallel strings of 125 series connected 3 000  $\mu\text{F}$ , 450 V electrolytic capacitors, in order to meet the voltage rating of 100 kV. Half of the DC bus capacitors are installed at the sending-end station and the other half at the receiving-end station. The DC bus will be robust against the effects of ripple current and the only other requirement is a reasonably stiff voltage on the DC side. Thus more parallel strings of series connected capacitors may be needed, although the DC line capacitance may be sufficient.

The converter generates characteristic harmonics related to the switching frequency. The AC currents are smoothed by the filter inductor and then the rest of the harmonic content of the AC voltage is reduced by the AC filter. The filter capacitors consist of 33 parallel strings of 83 series connected 50  $\mu\text{F}$ , 440 V capacitors in order to meet the voltage rating of 66 kV phase-to-phase. The equivalent rating of the filter capacitors is 10  $\mu\text{F}$  per phase, whereas the rating of the filter inductor depends on dynamic performance of the system and the ripple current limitations, which differs for each practical situation. The cost of the filter reactors was estimated as R100 000 each.

The cost of the DC bus LEMs is included in the cost for the control and measurement equipment. Auxiliary equipment costs include the cost of the DC line breakers, while other costs include transport costs, construction costs, field services, survey fees, engineering fees and general overhead costs. These were estimated as 50% of the total material cost.

For 8.5 MW long-distance transmission, with a  $\pm 50$  kV bipolar DC line, the cost for the two converter stations would amount to R1.8 million per MW. It is important to note that the values given in Table A-2 are estimated values and are subject to change.



## **ADDENDUM B PROGRAM LISTINGS**

### **B.1 OLP CODE**

B.1.1 Conductor.m

B.1.2 Tower.m

### **B.2 PSAT-RETICMASTER INTERFACE CODE**

B.2.1 Interface.m

### **B.3 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE CODE**

B.3.1 Device.m

### **B.4 TESAT CODE**

B.4.1 LFAC.m

B.4.2 LFDC.m

B.4.3 Plot2d.m

B.4.4 Plot3d.m

### **B.5 FUNCTION CODE TO CALCULATE THE THERMAL LOAD REACH OF THREE-PHASE LINES**

B.5.1 Constants.m

### **B.6 FUNCTION CODE TO CALCULATE THE TRANSFER CAPABILITIES OF THREE-PHASE LINES**

B.6.1 Transfer.m

## B.1 OLP CODE

### B.1.1 Conductor.m

```

1 function Conductor(action)
2 %Main Conductor GUI Function
3
4 switch(action)
5
6 case 'Apply_Settings'
7     %Display Selected Conductor Parameters
8     conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
9     value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');
10    set(findobj(findobj('Tag','OLPGUI'),'Tag','Stranding'),'String',conductor(value).Stranding);
11    set(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String',conductor(value).StrandDiam);
12    set(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String',conductor(value).OverDiam);
13    set(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String',conductor(value).R);
14    set(findobj(findobj('Tag','OLPGUI'),'Tag','Current'),'String',conductor(value).Current);
15
16    %Frequency of Operation
17    f=50;
18
19    %Get Bundle Selection, Conductor Parameters, Tower Parameters, Distance and Earth Resistivity
20    Bundleselect=get(findobj(findobj('Tag','OLPGUI'),'Tag','Conductors/Bundle'),'Value');
21    StrDia=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String'));
22    R=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String'));
23    Over_Diam=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String'));
24    Distance=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Distance'),'String'));
25    d=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'String'));
26    p=str2num(get(findobj(findobj('Tag','OLPGUI'),'Tag','p'),'String'));
27    Parameters=get(findobj('Tag','OLPGUI'),'UserData');
28
29    GMD=Parameters(5);
30    GMD0=Parameters(6);
31
32    %Calculations
33
34    %Calculation of the GMR of the conductor and the equivalent outside raduis (Dsc) of bundle selections
35    x=conductor(value).x;
36    if x==1.08955 %6/1
37        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
38        cla;
39        image(imread('ACSR6A1S.bmp','bmp'));
40        set(gca,'Tag','Axes1');
41        set(gca,'Visible','off');
42        set(gca,'Position',[921 541 68 62]);
43    elseif x==1.950552 %18/1
44        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
45        cla;
46        image(imread('ACSR18A1S.bmp','bmp'));
47        set(gca,'Tag','Axes1');
48        set(gca,'Visible','off');
49        set(gca,'Position',[914 541 82 82]);
50    elseif x==2.4118148 %24/7
51        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
52        cla;
53        image(imread('ACSR24A7S.bmp','bmp'));
54        set(gca,'Tag','Axes1');
55        set(gca,'Visible','off');
56        set(gca,'Position',[900 522 110 112]);
57    elseif x==2.56595 %26/7
58        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
59        cla;
60        image(imread('ACSR26A7S.bmp','bmp'));
61        set(gca,'Tag','Axes1');
62        set(gca,'Visible','off');
63        set(gca,'Position',[900 521 111 111]);
64    elseif x==2.892485 %30/7
65        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
66        cla;
67        image(imread('ACSR30A7S.bmp','bmp'));
68        set(gca,'Tag','Axes1');
69        set(gca,'Visible','off');
70        set(gca,'Position',[897 520 115 115]);
71    elseif x==3.174735 %45/7
72        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
73        cla;
74        %image(imread('ACSR45A7S.bmp','bmp'));
75        set(gca,'Tag','Axes1');
76        set(gca,'Visible','off');
77        set(gca,'Position',[889 516 131 128]);
78    elseif x==3.25136 %54/7
79        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
80        cla;
81        image(imread('ACSR54A7S.bmp','bmp'));
82        set(gca,'Tag','Axes1');
83        set(gca,'Visible','off');
84        set(gca,'Position',[889 516 131 128]);
85    else x==3.646384 %54/19
86        axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes1'))
87        cla;
88        image(imread('ACSR54A19S.bmp','bmp'));
89        set(gca,'Tag','Axes1');
90        set(gca,'Visible','off');
91        set(gca,'Position',[889 516 131 128]);

```



```

92     end;
93
94     Ds=(StrDia*x)/10;
95     if Bundleselect==1
96         GMR=Ds;
97         Dsc=Over_Diam/20;
98     elseif Bundleselect==2
99         GMR=sqrt(Ds*d);
100        Dsc=sqrt((Over_Diam/20)*d);
101     elseif Bundleselect==3
102        GMR=(Ds*d^2)^(1/3);
103        Dsc=((Over_Diam/20)*d^2)^(1/3);
104     elseif Bundleselect==4
105        GMR=1.091*((Ds*d^3)^(1/4));
106        Dsc=1.091*((Over_Diam/20)*d^3)^(1/4);
107     end;
108
109     %Calculation of R/km, L/km and C/km for positive and zero sequences
110     R=R/Bundleselect;
111     Re=9.869e-4*f;
112     De=(2160*0.3048)*(sqrt(p/f));
113     zs=(R+Re)+(2*pi*f^2e-4)*log(De/(GMR/100))*j;
114     zm=Re+(2*pi*f^2e-4)*log(De/GMD)*j;
115     z0=zs+2*zm;
116     z1=zs-zm;
117
118     R0=real(z0);
119     R1=real(z1);
120     L0=imag(z0)/(2*pi*f);
121     L1=imag(z1)/(2*pi*f);
122
123     GMR0=(Dsc/100)*(GMD^2);
124     Cap1=(2*pi*8.854e-9)/(log(GMD/(Dsc/100)));
125     Cap0=(2*pi*8.854e-9)/log(GMD0/GMR0);
126
127     y1=j*2*pi*f*Cap1;
128     y0=j*2*pi*f*Cap0;
129
130     %Calculation of characteristic impedance and propagation constant
131     Zc0=sqrt(z0/y0);
132     Prop_Const0=sqrt(z0*y0);
133
134     Zc1=sqrt(z1/y1);
135     Prop_Const1=sqrt(z1*y1);
136
137     %Calculation of ABCD parameters
138     A0=cosh(Prop_Const0*Distance);
139     D0=A0;
140     B0=Zc0*sinh(Prop_Const0*Distance);
141     C0=(1/Zc0)*sinh(Prop_Const0*Distance);
142
143     A1=cosh(Prop_Const1*Distance);
144     D1=A1;
145     B1=Zc1*sinh(Prop_Const1*Distance);
146     C1=(1/Zc1)*sinh(Prop_Const1*Distance);
147
148     %Calculation of Z' and Y'/2
149     Z1=B1;
150     Y1=tanh((Prop_Const1*Distance)/2)/Zc1;
151
152     Z0=B0;
153     Y0=tanh((Prop_Const0*Distance)/2)/Zc0;
154
155     %Store GMR, Distance, Dsc and StrDia in UserData field of MainDisplay figure
156     Parameters(1)=StrDia;
157     Parameters(2)=Dsc;
158     Parameters(3)=Distance;
159     Parameters(4)=GMR;
160     Parameters(7)=R;
161     Parameters(8)=p;
162
163     %Outputs
164     set(findobj(findobj('Tag','OLPGUI'),'Tag','GMRdisplay'),'String',sprintf('%3.4f,GMR));
165     set(findobj(findobj('Tag','OLPGUI'),'Tag','z1edit'),'String',num2str(z1,'%1.4g'));
166     set(findobj(findobj('Tag','OLPGUI'),'Tag','z0edit'),'String',num2str(z0,'%1.4g'));
167     set(findobj(findobj('Tag','OLPGUI'),'Tag','y1edit'),'String',num2str(y1,'%1.3g'));
168     set(findobj(findobj('Tag','OLPGUI'),'Tag','y0edit'),'String',num2str(y0,'%1.3g'));
169
170     set(findobj(findobj('Tag','OLPGUI'),'Tag','Z1edit'),'String',num2str(Z1,'%1.4g'));
171     set(findobj(findobj('Tag','OLPGUI'),'Tag','Y1edit'),'String',num2str(Y1,'%1.2e'));
172     set(findobj(findobj('Tag','OLPGUI'),'Tag','Z0edit'),'String',num2str(Z0,'%1.4g'));
173     set(findobj(findobj('Tag','OLPGUI'),'Tag','Y0edit'),'String',num2str(Y0,'%1.2e'));
174
175     set(findobj('Tag','OLPGUI'),'UserData',Parameters);
176
177     case 'BundleSelect'
178         value=get(findobj(findobj('Tag','OLPGUI'),'Tag','Conductors/Bundle'),'Value');
179
180         if value==1
181             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundletext'),'Enable','off');
182             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'Enable','off');
183             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'BackgroundColor',[.6941 .6941 .6941]);
184             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundlecmtxt'),'Enable','off');
185         else
186             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundletext'),'Enable','on');
187             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'Enable','on');
188             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundle Spacing'),'BackgroundColor',[1 1 1]);
189             set(findobj(findobj('Tag','OLPGUI'),'Tag','Bundlecmtxt'),'Enable','on');
190         end;
191
192     case 'Modify'

```

```

193 %Open Modify Conductor GUI with selected conductor parameters
194
195 fig = findobj('Tag','Modify Conductor');
196 if isempty(fig)
197     open('ModConductor.fig');
198 end
199
200 conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
201 value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');
202
203 set(findobj('Tag','Name'),'String',conductor(value).Name);
204 if conductor(value).x == 1.08955
205     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',1);
206     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
207     cla;
208     image(imread('ACSR6A1S.bmp','bmp'));
209     set(gca,'Tag','Axes1');
210     set(gca,'Color',[ 8 8 8])
211     set(gca,'Visible','off');
212     set(gca,'Position',[56 61 68 62]);
213 elseif conductor(value).x == 1.950552
214     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',2);
215     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
216     cla;
217     image(imread('ACSR18A1S.bmp','bmp'));
218     set(gca,'Tag','Axes1');
219     set(gca,'Color',[ 8 8 8])
220     set(gca,'Visible','off');
221     set(gca,'Position',[51 46 82 82]);
222 elseif conductor(value).x == 2.4118148
223     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',3);
224     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
225     cla;
226     image(imread('ACSR24A7S.bmp','bmp'));
227     set(gca,'Tag','Axes1');
228     set(gca,'Color',[ 8 8 8])
229     set(gca,'Visible','off');
230     set(gca,'Position',[36 36 110 112]);
231 elseif conductor(value).x == 2.56595
232     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',4);
233     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
234     cla;
235     image(imread('ACSR26A7S.bmp','bmp'));
236     set(gca,'Tag','Axes1');
237     set(gca,'Color',[ 8 8 8])
238     set(gca,'Visible','off');
239     set(gca,'Position',[36 36 111 111]);
240 elseif conductor(value).x == 2.892485
241     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',5);
242     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
243     cla;
244     image(imread('ACSR30A7S.bmp','bmp'));
245     set(gca,'Tag','Axes1');
246     set(gca,'Color',[ 8 8 8])
247     set(gca,'Visible','off');
248     set(gca,'Position',[36 31 115 115]);
249 elseif conductor(value).x == 3.174735
250     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',6);
251     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
252     cla;
253     %image(imread('ACSR???.bmp','bmp'));
254     set(gca,'Tag','Axes1');
255     set(gca,'Color',[ 8 8 8])
256     set(gca,'Visible','off');
257     set(gca,'Position',[51 46 109 109]);
258 elseif conductor(value).x == 3.25136
259     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',7);
260     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
261     cla;
262     image(imread('ACSR54A7S.bmp','bmp'));
263     set(gca,'Tag','Axes1');
264     set(gca,'Color',[ 8 8 8])
265     set(gca,'Visible','off');
266     set(gca,'Position',[26 26 131 128]);
267 else
268     set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandselect'),'Value',8);
269     axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
270     cla;
271     image(imread('ACSR54A19S.bmp','bmp'));
272     set(gca,'Tag','Axes1');
273     set(gca,'Color',[ 8 8 8])
274     set(gca,'Visible','off');
275     set(gca,'Position',[26 26 131 128]);
276 end;
277 set(findobj(findobj('Tag','Modify Conductor'),'Tag','Strand_Diam'),'String',conductor(value).StrandDiam);
278 set(findobj(findobj('Tag','Modify Conductor'),'Tag','Overall_Diam'),'String',conductor(value).OverDiam);
279 set(findobj(findobj('Tag','Modify Conductor'),'Tag','Rtext'),'String',conductor(value).R);
280 set(findobj(findobj('Tag','Modify Conductor'),'Tag','Current'),'String',conductor(value).Current);
281
282 case 'ModStranding'
283     Strandingselect=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value');
284     if Strandingselect==1
285         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
286         cla;
287         image(imread('ACSR6A1S.bmp','bmp'));
288         set(gca,'Tag','Axes1');
289         set(gca,'Color',[ 8 8 8])
290         set(gca,'Visible','off');
291         set(gca,'Position',[56 61 68 62]);
292     elseif Strandingselect==2
293         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))

```



```

294         cla;
295         image(imread('ACSR18A1S.bmp','bmp'));
296         set(gca,'Tag','Axes1');
297         set(gca,'Color',[ 8 8 8])
298         set(gca,'Visible','off');
299         set(gca,'Position',[51 46 82 82]);
300     elseif Strandingselect==3
301         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
302         cla;
303         image(imread('ACSR24A7S.bmp','bmp'));
304         set(gca,'Tag','Axes1');
305         set(gca,'Color',[ 8 8 8])
306         set(gca,'Visible','off');
307         set(gca,'Position',[36 36 110 112]);
308     elseif Strandingselect==4
309         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
310         cla;
311         image(imread('ACSR26A7S.bmp','bmp'));
312         set(gca,'Tag','Axes1');
313         set(gca,'Color',[ 8 8 8])
314         set(gca,'Visible','off');
315         set(gca,'Position',[36 36 111 111]);
316     elseif Strandingselect==5
317         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
318         cla;
319         image(imread('ACSR30A7S.bmp','bmp'));
320         set(gca,'Tag','Axes1');
321         set(gca,'Color',[ 8 8 8])
322         set(gca,'Visible','off');
323         set(gca,'Position',[36 31 115 115]);
324     elseif Strandingselect==6
325         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
326         cla;
327         %image(imread('ACSR???.bmp','bmp'));
328         set(gca,'Tag','Axes1');
329         set(gca,'Color',[ 8 8 8])
330         set(gca,'Visible','off');
331         set(gca,'Position',[51 46 109 109]);
332     elseif Strandingselect==7
333         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
334         cla;
335         image(imread('ACSR54A7S.bmp','bmp'));
336         set(gca,'Tag','Axes1');
337         set(gca,'Color',[ 8 8 8])
338         set(gca,'Visible','off');
339         set(gca,'Position',[26 26 131 128]);
340     else
341         axes(findobj(findobj('Tag','Modify Conductor'),'Tag','Axes1'))
342         cla;
343         image(imread('ACSR54A19S.bmp','bmp'));
344         set(gca,'Tag','Axes1');
345         set(gca,'Color',[ 8 8 8])
346         set(gca,'Visible','off');
347         set(gca,'Position',[26 26 131 128]);
348     end;
349
350 case 'ApplyModConductor'
351
352     %Get Conductor Structure
353     %Get Changed Conductor Parameters
354
355     conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
356     value=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'Value');
357
358     Name=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Name'),'String');
359     Strandingselect=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strandingselect'),'Value');
360     Strand_Diam=str2num(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Strand_Diam'),'String'));
361     Overall_Diam=get(findobj(findobj('Tag','Modify Conductor'),'Tag','Overall_Diam'),'String');
362     R=num2str(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Rtext'),'String'));
363     Current=num2str(get(findobj(findobj('Tag','Modify Conductor'),'Tag','Current'),'String'));
364
365     Stranding=0;
366     x=0;
367     [Stranding,x]=GetStranding(Strandingselect);
368
369     %Change Conductor Structure
370     conductor(value)=struct('Name',Name,'Stranding',Stranding,'StrandDiam',Strand_Diam,'OverDiam',Overall_Diam,
371                             'R',R,'Current',Current,'x',x);
372
373     for i=1:length(conductor)
374         CondSelect(i)={conductor(i).Name};
375     end
376
377     %Outputs
378     SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor);
379
380 case 'AddStranding'
381     Strandingselect=get(findobj(findobj('Tag','Add Conductor'),'Tag','Strandingselect'),'Value');
382     if Strandingselect==1
383         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
384         cla;
385         image(imread('ACSR6A1S.bmp','bmp'));
386         set(gca,'Tag','Axes1');
387         set(gca,'Color',[ 8 8 8])
388         set(gca,'Visible','off');
389         set(gca,'Position',[56 61 68 62]);
390     elseif Strandingselect==2
391         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
392         cla;
393         image(imread('ACSR18A1S.bmp','bmp'));
394         set(gca,'Tag','Axes1');

```

```

395         set(gca,'Color',[.8 .8 .8])
396         set(gca,'Visible','off');
397         set(gca,'Position',[51 46 82 82]);
398     elseif Strandingsselect==3
399         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
400         cla;
401         image(imread('ACSR24A7S.bmp','bmp'));
402         set(gca,'Tag','Axes1');
403         set(gca,'Color',[.8 .8 .8])
404         set(gca,'Visible','off');
405         set(gca,'Position',[36 36 110 112]);
406     elseif Strandingsselect==4
407         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
408         cla;
409         image(imread('ACSR26A7S.bmp','bmp'));
410         set(gca,'Tag','Axes1');
411         set(gca,'Color',[.8 .8 .8])
412         set(gca,'Visible','off');
413         set(gca,'Position',[36 36 111 111]);
414     elseif Strandingsselect==5
415         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
416         cla;
417         image(imread('ACSR30A7S.bmp','bmp'));
418         set(gca,'Tag','Axes1');
419         set(gca,'Color',[.8 .8 .8])
420         set(gca,'Visible','off');
421         set(gca,'Position',[36 31 115 115]);
422     elseif Strandingsselect==6
423         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
424         cla;
425         %image(imread('ACSR?? .bmp','bmp'));
426         set(gca,'Tag','Axes1');
427         set(gca,'Color',[.8 .8 .8])
428         set(gca,'Visible','off');
429         set(gca,'Position',[51 46 109 109]);
430     elseif Strandingsselect==7
431         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
432         cla;
433         image(imread('ACSR54A7S.bmp','bmp'));
434         set(gca,'Tag','Axes1');
435         set(gca,'Color',[.8 .8 .8])
436         set(gca,'Visible','off');
437         set(gca,'Position',[26 26 131 128]);
438     else
439         axes(findobj(findobj('Tag','Add Conductor'),'Tag','Axes1'))
440         cla;
441         image(imread('ACSR54A19S.bmp','bmp'));
442         set(gca,'Tag','Axes1');
443         set(gca,'Color',[.8 .8 .8])
444         set(gca,'Visible','off');
445         set(gca,'Position',[26 26 131 128]);
446     end;
447
448 case 'Add'
449
450     %Get Conductor Structure
451     %Get New Conductor Parameters
452     conductor=get(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData');
453
454     Name=get(findobj(findobj('Tag','Add Conductor'),'Tag','Name'),'String');
455     Strandingsselect=get(findobj(findobj('Tag','Add Conductor'),'Tag','Strandingsselect'),'Value');
456     Strand_Diam=str2num(get(findobj(findobj('Tag','Add Conductor'),'Tag','Strand_Diam'),'String'));
457     Overall_Diam=get(findobj(findobj('Tag','Add Conductor'),'Tag','Overall_Diam'),'String');
458     R=num2str(get(findobj(findobj('Tag','Add Conductor'),'Tag','Rtext'),'String'));
459     Current=num2str(get(findobj(findobj('Tag','Add Conductor'),'Tag','Current'),'String'));
460
461     Stranding=0;
462     x=0;
463     [Stranding,x]=GetStranding(Strandingsselect)
464
465     %Add New Conductor
466     sz=size(conductor);
467
468     conductor(sz(2)+1)=struct('Name',Name,'Stranding',Stranding,'StrandDiam',Strand_Diam,'OverDiam',Overall_Diam,'
469                             R',R,'Current',Current,'x',x);
470
471     for i=1:length(conductor)
472         CondSelect(i)={conductor(i).Name};
473     end
474
475     %Outputs
476     SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor);
477
478 case 'OpenAddConductor'
479     fig = findobj('Tag','Add Conductor');
480     if isempty(fig)
481         open('AddCond.fig');
482     end
483
484 %Close Add Conductor GUI
485 case 'CloseAddConductor'
486     fig = findobj('Tag','Add Conductor');
487     close(fgure(fig));
488
489 %Close Modify Conductor GUI
490 case 'CloseModConductor'
491     fig = findobj('Tag','Modify Conductor');
492     close(fgure(fig));
493
494 case 'CloseOLP'
495     fig = findobj('Tag','OLPGUI');

```



```

496     saveas(ffigure(fig),get(fig,'FileName'),'fig');
497     close(figure(fig));
498
499 end
500
501 %SubFunctions
502 function [Stranding,x]=GetStranding(Strandingselect)
503
504     if Strandingselect==1
505         Stranding='6/1';
506         x=1.08955;
507     elseif Strandingselect==2
508         Stranding='18/1';
509         x=1.950552;
510     elseif Strandingselect==3
511         Stranding='24/7';
512         x=2.4118148;
513     elseif Strandingselect==4
514         Stranding='26/7';
515         x=2.56595;
516     elseif Strandingselect==5
517         Stranding='30/7';
518         x=2.892485;
519     elseif Strandingselect==6
520         Stranding='45/7';
521         x=3.174735;
522     elseif Strandingselect==7
523         Stranding='54/7';
524         x=3.25136;
525     else
526         Stranding='54/19';
527         x=3.646384;
528     end,
529
530 function SetCondParameters(CondSelect,Stranding,Strand_Diam,Overall_Diam,R,Current,conductor)
531     set(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'String',CondSelect);
532     set(findobj(findobj('Tag','OLPGUI'),'Tag','Stranding'),'String',Stranding);
533     set(findobj(findobj('Tag','OLPGUI'),'Tag','Strand_Diam'),'String',Strand_Diam);
534     set(findobj(findobj('Tag','OLPGUI'),'Tag','Overall_Diam'),'String',Overall_Diam);
535     set(findobj(findobj('Tag','OLPGUI'),'Tag','Rtext'),'String',R);
536     set(findobj(findobj('Tag','OLPGUI'),'Tag','Current'),'String',Current);
537     set(findobj(findobj('Tag','OLPGUI'),'Tag','CondSelect'),'UserData',conductor);

```

## B.1.2 Tower.m

```

1 function Tower(action)
2 %Main Tower GUI Function
3
4 switch(action)
5
6 case 'TwoEarthWires'
7     set(gcbo,'Value',1);
8     set(findobj(findobj('Tag','OLPGUI'),Tag,'OneEarthRB'),'Value',0);
9     set(findobj(findobj('Tag','OLPGUI'),Tag,'NoEarthRB'),'Value',0);
10    set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'BackgroundColor',[1 1 1]);
11    set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'BackgroundColor',[1 1 1]);
12    set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'BackgroundColor',[1 1 1]);
13    set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'BackgroundColor',[1 1 1]);
14    set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'Enable','inactive');
15    set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'Enable','inactive');
16    set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'Enable','inactive');
17    set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'Enable','inactive');
18    set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1mtext'),'Enable','on');
19    set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2mtext'),'Enable','on');
20    set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1text'),'Enable','on');
21    set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2text'),'Enable','on');
22
23 case 'OneEarthWire'
24     set(gcbo,'Value',1);
25     set(findobj(findobj('Tag','OLPGUI'),Tag,'TwoEarthRB'),'Value',0);
26     set(findobj(findobj('Tag','OLPGUI'),Tag,'NoEarthRB'),'Value',0);
27     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'BackgroundColor',[1 1 1]);
28     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'BackgroundColor',[1 1 1]);
29     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
30     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
31     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'Enable','inactive');
32     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'Enable','inactive');
33     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'Enable','off');
34     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'Enable','off');
35     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1mtext'),'Enable','on');
36     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2mtext'),'Enable','off');
37     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1text'),'Enable','on');
38     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2text'),'Enable','off');
39
40 case 'NoEarthWires'
41     set(gcbo,'Value',1);
42     set(findobj(findobj('Tag','OLPGUI'),Tag,'TwoEarthRB'),'Value',0);
43     set(findobj(findobj('Tag','OLPGUI'),Tag,'OneEarthRB'),'Value',0);
44     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'BackgroundColor',[.6941 .6941 .6941]);
45     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'BackgroundColor',[.6941 .6941 .6941]);
46     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'BackgroundColor',[.6941 .6941 .6941]);
47     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'BackgroundColor',[.6941 .6941 .6941]);
48     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'Enable','off');
49     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'Enable','off');
50     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'Enable','off');
51     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'Enable','off');
52     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1mtext'),'Enable','off');
53     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2mtext'),'Enable','off');
54     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth1text'),'Enable','off');
55     set(findobj(findobj('Tag','OLPGUI'),Tag,'Earth2text'),'Enable','off');
56
57 case 'Apply_Settings'
58     %Set selected tower parameters
59     tower=get(findobj(findobj('Tag','OLPGUI'),Tag,'TowerSelect'),'UserData');
60     value=get(findobj(findobj('Tag','OLPGUI'),Tag,'TowerSelect'),'Value');
61
62     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horzaedit'),'String',tower(value).Horza);
63     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horzbedit'),'String',tower(value).Horzb);
64     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horzcedit'),'String',tower(value).Horzc);
65     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'String',tower(value).Horze1);
66     set(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'String',tower(value).Horze2);
67     set(findobj(findobj('Tag','OLPGUI'),Tag,'Vertaedit'),'String',tower(value).Verta);
68     set(findobj(findobj('Tag','OLPGUI'),Tag,'Vertbedit'),'String',tower(value).Vertb);
69     set(findobj(findobj('Tag','OLPGUI'),Tag,'Vertcedit'),'String',tower(value).Vertc);
70     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'String',tower(value).Verte1);
71     set(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'String',tower(value).Verte2);
72
73     %Frequency of operation
74     f=50;
75
76     %Get selected tower parameters
77     Horza=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Horzaedit'),'String'));
78     Horzb=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Horzbedit'),'String'));
79     Horzc=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Horzcedit'),'String'));
80     Horze1=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Horze1'),'String'));
81     Horze2=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Horze2'),'String'));
82     Verta=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Vertaedit'),'String'));
83     Vertb=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Vertbedit'),'String'));
84     Vertc=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Vertcedit'),'String'));
85     Verte1=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Verte1'),'String'));
86     Verte2=str2num(get(findobj(findobj('Tag','OLPGUI'),Tag,'Verte2'),'String'));
87
88     Parameters=get(findobj('Tag','OLPGUI'),'UserData');
89
90     GMR=Parameters(4);
91     Distance=Parameters(3);
92     Dsc=Parameters(2);
93     R=Parameters(7);
94     p=Parameters(8);
95
96

```



```

97 SetRB(Horze1,Verte1,Horze2,Verte2)
98 Name=tower(value).Name;
99 PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verb,Vertc,Verte1,Verte2)
100
101 %Calculations
102 %Calculation of R/km, L/km and C/km for positive and zero sequences
103 dab=sqrt((Verta-Verb)^2+(Horza-Horzb)^2);
104 dbc=sqrt((Verb-Vertc)^2+(Horzb-Horzc)^2);
105 dca=sqrt((Vertc-Verta)^2+(Horzc-Horza)^2);
106 GMD=(dab*dbc*dca)^(1/3);
107
108 dabi=sqrt((Verta+Verb)^2+(Horza-Horzb)^2);
109 daci=sqrt((Verta+Vertc)^2+(Horza-Horzc)^2);
110 dbci=sqrt((Verb+Vertc)^2+(Horzb-Horzc)^2);
111
112 GMD0=2*((Verta*Verb*Vertc)*(dabi*daci*dbci)^2)^(1/9);
113 GMR0=(Dsc/100)*(GMD^2);
114 Cap0=(2*pi*8.854e-9)/log(GMD0/GMR0);
115 Cap1=(2*pi*8.854e-9)/(log(GMD/(Dsc/100)));
116 y0=j*2*pi*f*Cap0;
117 y1=j*2*pi*f*Cap1;
118
119 Re=9.869e-4*f;
120 De=(2160*0.3048)*(sqrt(p/f));
121 zs=(R+Re)+(2*pi*f^2e-4)*log(De/(GMR/100))*j;
122 zm=Re+(2*pi*f^2e-4)*log(De/GMD)*j;
123 z0=zs+2*zm;
124 z1=zs-zm;
125
126 R0=real(z0);
127 R1=real(z1);
128 L0=imag(z0)/(2*pi*f);
129 L1=imag(z1)/(2*pi*f);
130
131 %Calculation of characteristic impedance and propagation constant
132 Zc0=sqrt(z0/y0);
133 Prop_Const0=sqrt(z0*y0);
134 Zc1=sqrt(z1/y1);
135 Prop_Const1=sqrt(z1*y1);
136
137 %Calculation of ABCD parameters
138 A0=cosh(Prop_Const0*Distance);
139 D0=A0;
140 B0=Zc0*sinh(Prop_Const0*Distance);
141 C0=(1/Zc0)*sinh(Prop_Const0*Distance);
142
143 A1=cosh(Prop_Const1*Distance);
144 D1=A1;
145 B1=Zc1*sinh(Prop_Const1*Distance);
146 C1=(1/Zc1)*sinh(Prop_Const1*Distance);
147
148 %Calculation of Z' and Y'/2
149 Z1=B1;
150 Y1=tanh((Prop_Const1*Distance)/2)/Zc1;
151
152 Z0=B0;
153 Y0=tanh((Prop_Const0*Distance)/2)/Zc0;
154
155 Parameters(5)=GMD;
156 Parameters(6)=GMD0;
157
158 %Outputs
159 set(findobj('Tag','OLPGUI'),'UserData',Parameters);
160 set(findobj(findobj('Tag','OLPGUI'),'Tag','GMDdisplay'),'String',sprintf('%3.4f,GMD));
161 set(findobj(findobj('Tag','OLPGUI'),'Tag','z1edit'),'String',num2str(z1,'%1.4g'));
162 set(findobj(findobj('Tag','OLPGUI'),'Tag','z0edit'),'String',num2str(z0,'%1.4g'));
163 set(findobj(findobj('Tag','OLPGUI'),'Tag','y1edit'),'String',num2str(y1,'%1.3g'));
164 set(findobj(findobj('Tag','OLPGUI'),'Tag','y0edit'),'String',num2str(y0,'%1.3g'));
165 set(findobj(findobj('Tag','OLPGUI'),'Tag','Z1edit'),'String',num2str(Z1,'%1.4g'));
166 set(findobj(findobj('Tag','OLPGUI'),'Tag','Y1edit'),'String',num2str(Y1,'%1.2e'));
167 set(findobj(findobj('Tag','OLPGUI'),'Tag','Z0edit'),'String',num2str(Z0,'%1.4g'));
168 set(findobj(findobj('Tag','OLPGUI'),'Tag','Y0edit'),'String',num2str(Y0,'%1.2e'));
169
170
171 case 'Add'
172 %Get Tower Structure
173
174 tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
175
176 %Get New Tower Parameters
177 Name=get(findobj(findobj('Tag','Add Tower'),'Tag','AddName'),'String');
178 Horza=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzaedit'),'String'));
179 Horzb=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzbedit'),'String'));
180 Horzc=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorzcedit'),'String'));
181 Verta=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVertaedit'),'String'));
182 Verbb=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVerbbedit'),'String'));
183 Vertc=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVertcedit'),'String'));
184 Verte2=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVerte2'),'String'));
185 Verte1=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddVerte1'),'String'));
186 Horze2=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorze2'),'String'));
187 Horze1=str2num(get(findobj(findobj('Tag','Add Tower'),'Tag','AddHorze1'),'String'));
188
189 %Add New Tower Structure
190 sz=size(tower);
191
192 tower(sz(2)+1)=struct('Name',Name,'Horza',Horza,'Horzb',Horzb,'Horzc',Horzc,'Horze1',Horze1,'Horze2',Horze2,
193 'Verta',Verta,'Verbb',Verbb,'Vertc',Vertc,'Verte1',Verte1,'Verte2',Verte2);
194
195 for i=1:length(tower)
196     TowerSelect(i)={tower(i).Name};
197 end

```

```

198
199     %Outputs
200     PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2);
201     PlotAddGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2)
202     SetRB(Horze1,Verte1,Horze2,Verte2)
203     SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2,tower);
204
205 case 'Modify'
206     %Open Modify Tower GUI with selected tower parameters
207
208     fig = findobj('Tag','Modify Tower');
209     if isempty(fig)
210         open('ModTower.fig');
211     end
212
213     tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
214     value=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'Value');
215
216     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModName'),'String',tower(value).Name);
217     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String',tower(value).Horza);
218     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String',tower(value).Horzb);
219     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String',tower(value).Horzc);
220     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String',tower(value).Verta);
221     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String',tower(value).Verbt);
222     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String',tower(value).Vertc);
223     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String',tower(value).Verte2);
224     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String',tower(value).Verte1);
225     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String',tower(value).Horze2);
226     set(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String',tower(value).Horze1);
227
228     Horza=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String'));
229     Horzb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String'));
230     Horzc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String'));
231     Verta=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String'));
232     Verbt=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String'));
233     Vertc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String'));
234     Verte2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String'));
235     Verte1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String'));
236     Horze2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String'));
237     Horze1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String'));
238
239     PlotModGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2)
240
241 case 'ApplyModTower'
242     %Get Tower Structure
243     tower=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData');
244     value=get(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'Value');
245
246     %Get Modified Tower Parameters
247     Name=get(findobj(findobj('Tag','Modify Tower'),'Tag','ModName'),'String');
248     Horza=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzaedit'),'String'));
249     Horzb=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzbedit'),'String'));
250     Horzc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorzcedit'),'String'));
251     Verta=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertaedit'),'String'));
252     Verbt=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertbedit'),'String'));
253     Vertc=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVertcedit'),'String'));
254     Verte2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte2'),'String'));
255     Verte1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModVerte1'),'String'));
256     Horze2=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze2'),'String'));
257     Horze1=str2num(get(findobj(findobj('Tag','Modify Tower'),'Tag','ModHorze1'),'String'));
258
259
260     tower(value)=struct('Name',Name,'Horza',Horza,'Horzb',Horzb,'Horzc',Horzc,'Horze1',Horze1,'Horze2',Horze2,
261         'Verta',Verta,'Verbt',Verbt,'Vertc',Vertc,'Verte1',Verte1,'Verte2',Verte2);
262
263     for i=1:length(tower)
264         TowerSelect(i)={tower(i).Name};
265     end
266
267     %Outputs
268     PlotOLPTower(Name,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2)
269     PlotModGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2)
270     SetRB(Horze1,Verte1,Horze2,Verte2)
271     SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2,tower);
272
273 case 'OpenAddTower'
274     fig = findobj('Tag','Add Tower');
275     if isempty(fig)
276         open('AddTower.fig');
277     end
278
279 %Close Add Tower GUI
280 case 'CloseAddTower'
281     fig = findobj('Tag','Add Tower');
282     close(figure(fig));
283
284 %Close Modify Tower GUI
285 case 'CloseModTower'
286     fig = findobj('Tag','Modify Tower');
287     close(figure(fig));
288
289 end
290
291 %SubFunction
292 function SetTowerParameters(TowerSelect,Horza,Horzb,Horzc,Horze1,Horze2,Verta,Verbt,Vertc,Verte1,Verte2,tower);
293     set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'String',TowerSelect);
294     set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData',tower);
295     set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzaedit'),'String',Horza);
296     set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzbedit'),'String',Horzb);
297     set(findobj(findobj('Tag','OLPGUI'),'Tag','Horzcedit'),'String',Horzc);
298     set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'String',Horze1);

```



```

299     set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'String',Horze2);
300     set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertaedit'),'String',Verta);
301     set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertedit'),'String',Vertb);
302     set(findobj(findobj('Tag','OLPGUI'),'Tag','Vertedit'),'String',Vertc);
303     set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'String',Verte1);
304     set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'String',Verte2);
305     set(findobj(findobj('Tag','OLPGUI'),'Tag','TowerSelect'),'UserData',tower);
306
307 function PlotOLTower(Name,Horza,HorzH,HorzC,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
308     oneearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value');
309     twoearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value');
310     xmax=0;
311     ymax=0;
312
313     axes(findobj(findobj('Tag','OLPGUI'),'Tag','Axes2'))
314     cla;
315     plot(Horza,Verta,'r o');
316     hold on;
317     set(gca,'Tag','Axes2');
318     if Horza>xmax
319         xmax=Horza;
320     end;
321     if Vertc>ymax
322         ymax=Vertc;
323     end;
324
325     xmax=Horza;
326     ymax=Verta;
327     set(gca,'Tag','Axes2');
328     plot(HorzH,Vertb,'w o');
329     hold on;
330     set(gca,'Tag','Axes2');
331     if HorzH>xmax
332         xmax=HorzH;
333     end;
334     if Vertb>ymax
335         ymax=Vertb;
336     end;
337
338     plot(HorzC,Vertc,'b o');
339     hold on;
340     set(gca,'Tag','Axes2');
341     if HorzC>xmax
342         xmax=HorzC;
343     end;
344     if Vertc>ymax
345         ymax=Vertc;
346     end;
347
348     if (oneearth==1) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
349         plot(Horze1,Verte1,'k o');
350         if Horze1>xmax
351             xmax=Horze1;
352         end;
353         if Verte1>ymax
354             ymax=Verte1;
355         end;
356         hold on;
357         set(gca,'Tag','Axes2');
358     elseif twoearth==1 & (isempty(Horze1)==0) & (isempty(Verte1)==0) & (isempty(Horze2)==0) & (isempty(Verte2)==0)
359         plot(Horze1,Verte1,'k o');
360         if Horze1>xmax
361             xmax=Horze1;
362         end;
363         if Verte1>ymax
364             ymax=Verte1;
365         end;
366         hold on;
367         set(gca,'Tag','Axes2');
368         plot(Horze2,Verte2,'k o');
369         if Horze2>xmax
370             xmax=Horze2;
371         end;
372         if Verte2>ymax
373             ymax=Verte2;
374         end;
375         hold on;
376         set(gca,'Tag','Axes2');
377     end;
378
379     x=0;
380     y=0.025*(ymax+1);
381     plot(x,y,'k');
382     set(gca,'Tag','Axes2');
383
384     set(gca,'Position',[842 61 170 310]);
385     set(gca,'Color',[.6941 .6941 .6941]);
386     axis([-2*xmax+1] (2*xmax+1) 0 (1.25*ymax+1)));
387     ylabel('Vertical Spacing [m]', 'FontSize',10);
388     xlabel('Horizontal Spacing [m]', 'FontSize',10);
389     legend('phase a', 'phase b', 'phase c');
390     title(Name, 'FontSize', 10);
391
392     hold off;
393
394 function PlotModGUITower(Horza,HorzH,HorzC,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
395     xmax=0;
396     ymax=0;
397
398     axes(findobj(findobj('Tag','Modify Tower'),'Tag','ModGUITowerAxes'))
399     cla;

```

```

400     plot(Horza, Verta, 'r o');
401     if Horza>xmax
402         xmax=Horza;
403     end;
404     if Vertb>ymax
405         ymax=Verta;
406     end;
407     hold on;
408     set(gca,'Tag','ModGUITowerAxes');
409
410     plot(Horz, Vertb, 'w o');
411     if Horzb>xmax
412         xmax=Horzb;
413     end;
414     if Vertb>ymax
415         ymax=Vertb;
416     end;
417     hold on;
418     set(gca,'Tag','ModGUITowerAxes');
419
420     plot(Horz, Vertc, 'b o');
421     if Horzc>xmax
422         xmax=Horzc;
423     end;
424     if Vertc>ymax
425         ymax=Vertc;
426     end;
427     hold on;
428     set(gca,'Tag','ModGUITowerAxes');
429
430     plot(Horze1, Verte1, 'k o');
431     if Horze1>xmax
432         xmax=Horze1;
433     end;
434     if Verte1>ymax
435         ymax=Verte1;
436     end;
437     hold on;
438     set(gca,'Tag','ModGUITowerAxes');
439
440     plot(Horze2, Verte2, 'k o');
441     if Horze2>xmax
442         xmax=Horze2;
443     end;
444     if Verte2>ymax
445         ymax=Verte2;
446     end;
447     hold on;
448     set(gca,'Tag','ModGUITowerAxes');
449
450     x=0;
451     y=0.025*(ymax+1);
452     plot(x,y, 'k');
453     set(gca,'Tag','ModGUITowerAxes');
454
455     set(gca,'Position',[56 56 120 195]);
456     set(gca,'Color',[ 8 8 8]);
457     axis([-2*xmax+1] (2*xmax+1) 0 (1.25*ymax+1))););
458     ylabel('Vertical Spacing [m]', 'FontSize',10);
459     xlabel('Horizontal Spacing [m]', 'FontSize',10);
460     hold off;
461
462 function PlotAddGUITower(Horza,Horzb,Horzc,Horze1,Horze2,Verta,Vertb,Vertc,Verte1,Verte2)
463     xmax=0;
464     ymax=0;
465
466     axes(findobj(findobj('Tag','Add Tower'),'Tag','AddGUITowerAxes'))
467     cla;
468     plot(Horza, Verta, 'r o');
469     if Horza>xmax
470         xmax=Horza;
471     end;
472     if Vertb>ymax
473         ymax=Verta;
474     end;
475     hold on;
476     set(gca,'Tag','AddGUITowerAxes');
477
478     plot(Horz, Vertb, 'w o');
479     if Horzb>xmax
480         xmax=Horzb;
481     end;
482     if Vertb>ymax
483         ymax=Vertb;
484     end;
485     hold on;
486     set(gca,'Tag','AddGUITowerAxes');
487
488     plot(Horz, Vertc, 'b o');
489     if Horzc>xmax
490         xmax=Horzc;
491     end;
492     if Vertc>ymax
493         ymax=Vertc;
494     end;
495     hold on;
496     set(gca,'Tag','AddGUITowerAxes');
497
498     plot(Horze1, Verte1, 'k o');
499     if Horze1>xmax
500         xmax=Horze1;

```



```

501     end;
502     if Vertel1>ymax
503         ymax=Vertel1;
504     end;
505     hold on;
506     set(gca,'Tag','AddGUITowerAxes');
507
508     plot(Horze2,Verte2,'k o');
509     if Horze2>xmax
510         xmax=Horze2;
511     end;
512     if Verte2>ymax
513         ymax=Verte2;
514     end;
515     hold on;
516     set(gca,'Tag','AddGUITowerAxes');
517
518     x=0;
519     y=0:0.025*(ymax+1);
520     plot(x,y,'k');
521     set(gca,'Tag','AddGUITowerAxes');
522
523     set(gca,'Position',[56 56 120 195]);
524     set(gca,'Color',[ 8 8 8]);
525     axis([-2*xmax+1] (2*xmax+1) 0 (1.25*ymax+1)));
526     ylabel('Vertical Spacing [m]', 'FontSize',10);
527     xlabel('Horizontal Spacing [m]', 'FontSize',10);
528     hold off;
529
530 function SetRB(Horze1,Verte1,Horze2,Verte2)
531     if (isempty(Horze2)==1) & (isempty(Verte2)==1) & (isempty(Horze1)==1) & (isempty(Verte1)==1)
532         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
533         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','off');
534         set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value',0);
535         set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','off');
536         set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',1);
537         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
538         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[ 6941 6941 6941]);
539         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[ 6941 6941 6941]);
540         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[ 6941 6941 6941]);
541         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[ 6941 6941 6941]);
542         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','off');
543         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','off');
544         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
545         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
546         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','off');
547         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
548         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','off');
549         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');
550     elseif (isempty(Horze2)==1) & (isempty(Verte2)==1) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
551         oneearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Value');
552         noearth=get(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value');
553         if (oneearth==0) & (noearth==0)
554             set(findobj(findobj('Tag','OLPGUI'),'Tag','NoEarthRB'),'Value',1);
555             set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'BackgroundColor',[ 6941 6941 6941]);
556             set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'BackgroundColor',[ 6941 6941 6941]);
557             set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte1'),'Enable','off');
558             set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze1'),'Enable','off');
559             set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1mtext'),'Enable','off');
560             set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth1text'),'Enable','on');
561         end;
562         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Value',0);
563         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','off');
564         set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','on');
565         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'BackgroundColor',[ 6941 6941 6941]);
566         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'BackgroundColor',[ 6941 6941 6941]);
567         set(findobj(findobj('Tag','OLPGUI'),'Tag','Verte2'),'Enable','off');
568         set(findobj(findobj('Tag','OLPGUI'),'Tag','Horze2'),'Enable','off');
569         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2mtext'),'Enable','off');
570         set(findobj(findobj('Tag','OLPGUI'),'Tag','Earth2text'),'Enable','off');
571     elseif (isempty(Horze2)==0) & (isempty(Verte2)==0) & (isempty(Horze1)==0) & (isempty(Verte1)==0)
572         set(findobj(findobj('Tag','OLPGUI'),'Tag','TwoEarthRB'),'Enable','on');
573         set(findobj(findobj('Tag','OLPGUI'),'Tag','OneEarthRB'),'Enable','on');
574     end;

```

## B.2 PSAT-RETICMASTER INTERFACE CODE

### B.2.1 *Interface.m*

```

1 function Interface(action)
2
3 switch(action)
4
5 case 'Source'
6     set(gcbo,'Value',1);
7     set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherRB'),'Value',0);
8     set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'Enable','off');
9     set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'BackgroundColor',[.6941 .6941 .6941]);
10    set(findobj(findobj('Tag','PRIGUI'),'Tag','Sources'),'Enable','on');
11    set(findobj(findobj('Tag','PRIGUI'),'Tag','Sources'),'BackgroundColor',[1 1 1]);
12    set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceVbase'),'Enable','on');
13    set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceVbase'),'BackgroundColor',[1 1 1]);
14    set(findobj(findobj('Tag','PRIGUI'),'Tag','kVtext'),'Enable','on');
15
16 case 'Other'
17     set(gcbo,'Value',1);
18     set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceRB'),'Value',0);
19     set(findobj(findobj('Tag','PRIGUI'),'Tag','Sources'),'Enable','off');
20     set(findobj(findobj('Tag','PRIGUI'),'Tag','Sources'),'BackgroundColor',[.6941 .6941 .6941]);
21     set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceVbase'),'Enable','off');
22     set(findobj(findobj('Tag','PRIGUI'),'Tag','SourceVbase'),'BackgroundColor',[.6941 .6941 .6941]);
23     set(findobj(findobj('Tag','PRIGUI'),'Tag','kVtext'),'Enable','off');
24     set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'Enable','on');
25     set(findobj(findobj('Tag','PRIGUI'),'Tag','OtherListbox'),'BackgroundColor',[1 1 1]);
26
27 case 'LoadData'
28     filename=get(findobj('Tag','filename'),'String');
29     fid=fopen(filename,'rt');
30     data=fgetl(fid);
31     data=fgetl(fid);
32     i=1;
33
34     while feof(fid) == 0
35         data=fgetl(fid);
36         name=fscanf(fid,'%s',1);
37         Imagnitude=fscanf(fid,'%g',2);
38         status=fseek(fid,1,'cof');
39         langle=fscanf(fid,'%g',1);
40         status=fseek(fid,1,'cof');
41         Vmagnitude=fscanf(fid,'%g',1);
42         status=fseek(fid,2,'cof');
43         Vangle=fscanf(fid,'%g',1);
44         status=fseek(fid,1,'cof');
45         Source=fscanf(fid,'%s',1);
46         status=fseek(fid,6,'cof');
47         Vbase=fscanf(fid,'%g',1);
48         Vunit=fscanf(fid,'%s',1);
49         status=fseek(fid,1,'cof');
50         technology=fscanf(fid,'%s',1);
51
52         matchsource=strcmp(Source,'Source');
53         if matchsource == 1
54             set(findobj('Tag','Sources'),'String',name);
55             sourcesr(1)=Vmagnitude;
56             sourcesr(2)=Vangle;
57             sourcesr(3)=Imagnitude;
58             sourcesr(4)=langle;
59         else
60             retic(i).name=name;
61             retic(i).Imagnitude=Imagnitude;
62             retic(i).langle=langle;
63             retic(i).Vmagnitude=Vmagnitude;
64             retic(i).Vangle=Vangle;
65             matchbase=strcmp(Vunit,'kV');
66             matchtech=strcmp(technology,'Single');
67             if (matchtech == 0) & (matchbase == 1)
68                 Vbase=Vbase*1000;
69             elseif (matchtech == 1) & (matchbase == 1)
70                 Vbase=sqrt(3)*Vbase*1000;
71             elseif (matchtech == 1) & (matchbase == 0)
72                 Vbase=sqrt(3)*Vbase;
73             end;
74             retic(i).Vbase=Vbase;
75             i=i+1;
76         end;
77     end;
78
79     fclose(fid);
80
81     for k=1:(length(retic)-1)
82         NameSelect(k)={retic(k).name};
83     end
84
85     set(findobj('Tag','Sources'),'UserData',sourcesr);
86     set(findobj('Tag','LoadListbox'),'Value',1);
87     set(findobj('Tag','OtherListbox'),'Value',1);
88     set(findobj('Tag','LoadListbox'),'String',NameSelect);
89     set(findobj('Tag','OtherListbox'),'String',NameSelect);
90     set(findobj('Tag','PRIGUI'),'UserData',retic);
91

```



```

92 case 'Apply_Settings'
93     SourceRB=get(findobj(findobj('Tag','PRIGUI'),'Tag','SourceRB'),'Value');
94     OtherRB=get(findobj(findobj('Tag','PRIGUI'),'Tag','OtherRB'),'Value');
95
96     SBASE=100e6;
97     retic=get(findobj('Tag','PRIGUI'),'UserData');
98     value=get(findobj('Tag','LoadListbox'),'Value');
99     valuesr=get(findobj('Tag','OtherListbox'),'Value');
100
101     %Calculate load end voltage (p.u.) (phase a)
102     Vloadmagnitude=(retic(value).Vmagnitude)/100;
103     Vloadangle=(retic(value).Vangle*pi)/180;
104     Vload=Vloadmagnitude*cos(Vloadangle)+j*Vloadmagnitude*sin(Vloadangle);
105
106     %Calculate load end current (p.u.) (phase a)
107     Iloadmagnitude=(sqrt(3)*(retic(value).Imagnitude*retic(value).Vbase))/SBASE;
108     Iloadangle=(retic(value).Iangle*pi)/180;
109     Iload=Iloadmagnitude*cos(Iloadangle)+j*Iloadmagnitude*sin(Iloadangle);
110
111     %Set outputs
112     set(findobj('Tag','Vloadmagnitude'),'String',Vloadmagnitude);
113     set(findobj('Tag','Vloadangle'),'String',sprintf('%3.2f',retic(value).Vangle));
114
115     %-----
116
117     if SourceRB==1
118         sourcesr=get(findobj('Tag','Sourcesr'),'UserData');
119         SourcesrVbase=str2num(get(findobj('Tag','SourcesrVbase'),'String'));
120
121         %Calculate Sending-End or Receiving-End source voltage (p.u.) (phase a)
122         Vsrmagnitude=sourcesr(1)/100;
123         Vsrange=(sourcesr(2)*pi)/180;
124         Vsr=Vsrmagnitude*cos(Vsrange)+j*Vsrmagnitude*sin(Vsrange);
125
126         %Calculate Sending-End or Receiving-End source current (p.u.) (phase a)
127         Isrmagnitude=(sqrt(3)*sourcesr(3)*(SourcesrVbase*1000))/SBASE;
128         Isrange=(sourcesr(4)*pi)/180;
129         Isr=Isrmagnitude*cos(Isrange)+j*Isrmagnitude*sin(Isrange);
130
131     elseif OtherRB==1
132         %Calculate Sending-End or Receiving-End source voltage (p.u.) (phase a)
133         Vsrmagnitude=(retic(valuesr).Vmagnitude)/100;
134         Vsrange=(retic(valuesr).Vangle*pi)/180;
135         Vsr=Vsrmagnitude*cos(Vsrange)+j*Vsrmagnitude*sin(Vsrange);
136
137         %Calculate Sending-End or Receiving-End source current (p.u.) (phase a)
138         Isrmagnitude=(sqrt(3)*(retic(valuesr).Imagnitude*retic(valuesr).Vbase))/SBASE;
139         Isrange=(retic(valuesr).Iangle*pi)/180;
140         Isr=Isrmagnitude*cos(Isrange)+j*Isrmagnitude*sin(Isrange);
141     end;
142
143     %-----
144
145     %Calculation of Zshunt, Zseries, Is or Ir, Zs or Zr and Vs or Vr
146
147     Zshunt=Vload/(Isr-Iload);
148     Zseries=(Vsr-Vload)/Isr;
149     Zeq=(Zseries*Zshunt)/(Zseries+Zshunt);
150     Veq=Zeq*(Vsr/Zseries);
151     Ieq=(Veq-Vload)/Zeq;
152
153     Veqmagnitude = abs(Veq); % magnitude
154     Veqangle = (180*angle(Veq))/pi; % phase angle
155     Ieqmagnitude = abs(Ieq); % magnitude
156     Ieqangle = (180*angle(Ieq))/pi; % phase angle
157
158     %Set outputs
159     set(findobj('Tag','Veqmagnitude'),'String',sprintf('%3.4f',Veqmagnitude));
160     set(findobj('Tag','Veqangle'),'String',sprintf('%3.2f',Veqangle));
161     set(findobj('Tag','Ieqmagnitude'),'String',sprintf('%3.2e',Ieqmagnitude));
162     set(findobj('Tag','Ieqangle'),'String',sprintf('%3.2f',Ieqangle));
163     set(findobj('Tag','Zeq'),'String',num2str(Zeq,'%3.2e'));
164
165 case 'ClosePRI'
166     fig = findobj('Tag','PRIGUI');
167     saveas(fig,'get(fig, 'FileName')', 'fig');
168     close(fig);
169
170 end;

```

## B.3 SUPPORT TECHNOLOGY AND DISPERSED GENERATION DATABASE CODE

### B.3.1 *Device.m*

```

1 function Device(action)
2 %Main Device GUI Function
3
4 switch(action)
5
6 case 'DGAddGUI'
7     set(gcbo,'Value',1);
8     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
9     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
10    set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
11    set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
12 case 'TransistorAddGUI'
13     set(gcbo,'Value',1);
14     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
15     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
16     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
17     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
18 case 'ThyristorAddGUI'
19     set(gcbo,'Value',1);
20     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
21     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
22     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
23     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
24 case 'MechanicalAddGUI'
25     set(gcbo,'Value',1);
26     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
27     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
28     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value',0);
29     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
30 case 'PassiveAddGUI'
31     set(gcbo,'Value',1);
32     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
33     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
34     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value',0);
35     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value',0);
36 case 'DGModGUI'
37     set(gcbo,'Value',1);
38     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
39     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
40     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
41     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
42 case 'TransistorModGUI'
43     set(gcbo,'Value',1);
44     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
45     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
46     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
47     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
48 case 'ThyristorModGUI'
49     set(gcbo,'Value',1);
50     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
51     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
52     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
53     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
54 case 'MechanicalModGUI'
55     set(gcbo,'Value',1);
56     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
57     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
58     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
59     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
60 case 'PassiveModGUI'
61     set(gcbo,'Value',1);
62     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
63     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
64     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
65     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
66 case 'DGDeviceGUI'
67     set(gcbo,'Value',1);
68     set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
69     set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
70     set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
71     set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
72     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
73     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[1 1 1]);
74 case 'TransistorDeviceGUI'
75     set(gcbo,'Value',1);
76     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
77     set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
78     set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
79     set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
80     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[1 1 1]);
81     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
82 case 'ThyristorDeviceGUI'
83     set(gcbo,'Value',1);
84     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
85     set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
86     set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
87     set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);

```



```

88     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
89     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
90 case 'MechanicalDeviceGUI'
91     set(gcbo,'Value',1);
92     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
93     set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
94     set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
95     set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value',0);
96     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
97     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
98 case 'PassiveDeviceGUI'
99     set(gcbo,'Value',1);
100    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value',0);
101    set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value',0);
102    set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value',0);
103    set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value',0);
104    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
105    set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGDeviceControl'),'BackgroundColor',[.6941 .6941 .6941]);
106 case 'SinglephaseAddGUI'
107     set(gcbo,'Value',1);
108     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','3phaseRB'),'Value',0);
109 case 'ThreephaseAddGUI'
110     set(gcbo,'Value',1);
111     set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','1phaseRB'),'Value',0);
112 case 'SinglephaseModGUI'
113     set(gcbo,'Value',1);
114     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3phaseRB'),'Value',0);
115 case 'ThreephaseModGUI'
116     set(gcbo,'Value',1);
117     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1phaseRB'),'Value',0);
118 case 'SinglephaseDeviceGUI'
119     set(gcbo,'Value',1);
120     set(findobj(findobj('Tag','DeviceGUI'),'Tag','3phaseRB'),'Value',0);
121 case 'ThreephaseDeviceGUI'
122     set(gcbo,'Value',1);
123     set(findobj(findobj('Tag','DeviceGUI'),'Tag','1phaseRB'),'Value',0);
124
125 case 'Calculate'
126     Sbase=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Sbase'),'String'));
127     Vbase=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Vbase'),'String'));
128     Ppu=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Ppu'),'String'));
129     Qpu=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Qpu'),'String'));
130
131     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value',1);
132     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String','');
133
134     if isempty(Ppu) & ~isempty(Qpu)
135         Q=Qpu*Sbase;
136         S=Q;
137         set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String,num2str(S));
138         set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String,num2str(Q));
139         set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String','');
140     elseif ~isempty(Ppu) & ~isempty(Qpu)
141         P=Ppu*Sbase;
142         Q=Qpu*Sbase;
143         S=abs(P+j*Q);
144         set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String,num2str(S));
145         set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String,num2str(P));
146         set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String,num2str(Q));
147     elseif ~isempty(Ppu) & isempty(Qpu)
148         S=Ppu*Sbase;
149         P=S;
150         set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String,num2str(S));
151         set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String,num2str(P));
152         set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String','');
153     else isempty(Ppu) & isempty(Qpu)
154         set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String','');
155         set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String','');
156         set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String','');
157     end;
158
159     set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String,num2str(Vbase));
160     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String','fit device');
161
162 case 'Display_Device'
163     %Display Selected Conductor Parameters
164     DDData=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'UserData');
165
166     value=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value');
167
168     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String',DDData(value).Info);
169
170     set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',DDData(value).Voltage);
171     set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',DDData(value).P);
172     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',DDData(value).Q);
173     P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
174     Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
175     S=abs(P+j*Q);
176     set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String,num2str(S));
177
178 case 'Fit_Device'
179     ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
180     SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
181     SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
182     ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
183
184     DeviceConfig=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceConfig'),'String');
185     V=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String'));
186     P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
187     Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
188     EF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','EnergyStorageCB'),'Value');

```

```

189 SF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','1phaseRB'),'Value');
190 TF=get(findobj(findobj('Tag','DeviceGUI'),'Tag','3phaseRB'),'Value');
191 Passive=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'Value');
192 Mechanical=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'Value');
193 Thyristor=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'Value');
194 Transistor=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'Value');
195 DG=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'Value');
196
197 %Initialize Variables
198 k=1;
199 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value',1);
200 Devices={};
201 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
202
203 %Find Devices
204 %Shunt Configuration
205 if (strcmp(DeviceConfig,'Shunt')==1);
206     for i=1:length(ShuntStructure)
207         if isempty(V)|V>str2num(ShuntStructure(i).Voltage)
208             VF=0;
209         else
210             VF=1;
211         end;
212         if isempty(P)|P>str2num(ShuntStructure(i).P)
213             PF=0;
214         else
215             PF=1;
216         end;
217         if isempty(Q)|Q>str2num(ShuntStructure(i).Q)
218             QF=0;
219         else
220             QF=1;
221         end;
222         A=[VF PF QF EF SF TF];
223         B=[ShuntStructure(i).VF ShuntStructure(i).PF ShuntStructure(i).QF ShuntStructure(i).
224             Energy ShuntStructure(i).Singlephase ShuntStructure(i).Threephase];
225         if (ShuntStructure(i).DeviceTech==1) & (A==B) & (Passive==1);           Shunt-Passive
226             DData(k)=ShuntStructure(i);
227             for l=1:length(DData)
228                 Devices(l)={DData(l).Name};
229                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
230             end
231             k=k+1;
232         elseif (ShuntStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);      %Shunt-Mechanical
233             DData(k)=ShuntStructure(i);
234             for l=1:length(DData)
235                 Devices(l)={DData(l).Name};
236                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
237             end
238             k=k+1;
239         elseif (ShuntStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);      %Shunt-Thyristor
240             DData(k)=ShuntStructure(i);
241             for l=1:length(DData)
242                 Devices(l)={DData(l).Name};
243                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
244             end
245             k=k+1;
246         elseif (ShuntStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);    %Shunt-In-line
247             DData(k)=ShuntStructure(i);
248             for l=1:length(DData)
249                 Devices(l)={DData(l).Name};
250                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
251             end
252             k=k+1;
253         elseif (ShuntStructure(i).DeviceTech==5) & (A==B) & (DG==1);            %Distributed Generation
254             DData(k)=ShuntStructure(i);
255             for l=1:length(DData)
256                 Devices(l)={DData(l).Name};
257                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
258             end
259             k=k+1;
260         elseif isempty(Devices)
261             Devices={};
262             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
263         end;
264     end;
265
266 %Series Configuration
267 elseif (strcmp(DeviceConfig,'Series')==1);
268     for i=1:length(SeriesStructure)
269         if isempty(V)|V>str2num(SeriesStructure(i).Voltage)
270             VF=0;
271         else
272             VF=1;
273         end;
274         if isempty(P)|P>str2num(SeriesStructure(i).P)
275             PF=0;
276         else
277             PF=1;
278         end;
279         if isempty(Q)|Q>str2num(SeriesStructure(i).Q)
280             QF=0;
281         else
282             QF=1;
283         end;
284         A=[VF PF QF EF SF TF];
285         B=[SeriesStructure(i).VF SeriesStructure(i).PF SeriesStructure(i).QF SeriesStructure(i).
286             Energy SeriesStructure(i).Singlephase SeriesStructure(i).Threephase];
287         if (SeriesStructure(i).DeviceTech==1) & (A==B) & (Passive==1);           %Series-Passive
288             DData(k)=SeriesStructure(i);
289             for l=1:length(DData)

```



```

290         Devices(l)={DData(l).Name};
291         set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
292     end
293     k=k+1;
294     elseif (SeriesStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);           %Series-Mechanical
295         DData(k)=SeriesStructure(i);
296         for l=1:length(DData)
297             Devices(l)={DData(l).Name};
298             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
299         end
300         k=k+1;
301     elseif (SeriesStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);           %Series-Thyristor
302         DData(k)=SeriesStructure(i);
303         for l=1:length(DData)
304             Devices(l)={DData(l).Name};
305             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
306         end
307         k=k+1;
308     elseif (SeriesStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);           %Series-Transistor
309         DData(k)=SeriesStructure(i);
310         for l=1:length(DData)
311             Devices(l)={DData(l).Name};
312             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
313         end
314         k=k+1;
315     elseif isempty(Devices)
316         Devices={};
317         set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
318     end;
319 end;
320
321 %Series-Shunt Configuration
322 elseif (strcmp(DeviceConfig,'Series-shunt')==1);
323     for i=1:length(SSStructure)
324         if isempty(V)|V>str2num(SSStructure(i).Voltage)
325             VF=0;
326         else
327             VF=1;
328         end;
329         if isempty(P)|P>str2num(SSStructure(i).P)
330             PF=0;
331         else
332             PF=1;
333         end;
334         if isempty(Q)|Q>str2num(SSStructure(i).Q)
335             QF=0;
336         else
337             QF=1;
338         end;
339         A=[VF PF QF EF SF TF];
340         B=[SSStructure(i).VF SSStructure(i).PF SSStructure(i).QF SSStructure(i).Energy
341           SSStructure(i).Singlephase SSStructure(i).Threephase];
342         if (SSStructure(i).DeviceTech==1) & (A==B) & (Passive==1);           %Series-shunt-Passive
343             DData(k)=SSStructure(i);
344             for l=1:length(DData)
345                 Devices(l)={DData(l).Name};
346                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
347             end
348             k=k+1;
349         elseif (SSStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);           %Series-shunt-Mechanical
350             DData(k)=SSStructure(i);
351             for l=1:length(DData)
352                 Devices(l)={DData(l).Name};
353                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
354             end
355             k=k+1;
356         elseif (SSStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);           %Series-shunt-Thyristor
357             DData(k)=SSStructure(i);
358             for l=1:length(DData)
359                 Devices(l)={DData(l).Name};
360                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
361             end
362             k=k+1;
363         elseif (SSStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);           %Series-shunt-Transistor
364             DData(k)=SSStructure(i);
365             for l=1:length(DData)
366                 Devices(l)={DData(l).Name};
367                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
368             end
369             k=k+1;
370         elseif isempty(Devices)
371             Devices={};
372             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
373         end;
374     end;
375
376 %In-line Configuration
377 elseif (strcmp(DeviceConfig,'In-line')==1);
378     for i=1:length(ILStructure)
379         if isempty(V)|V>str2num(ILStructure(i).Voltage)
380             VF=0;
381         else
382             VF=1;
383         end;
384         if isempty(P)|P>str2num(ILStructure(i).P)
385             PF=0;
386         else
387             PF=1;
388         end;
389         if isempty(Q)|Q>str2num(ILStructure(i).Q)
390             QF=0;

```

```

391         else
392             QF=1;
393         end;
394         A=[VF PF QF EF SF TF];
395         B=[ILStructure(i).VF ILStructure(i).PF ILStructure(i).QF ILStructure(i).
396             Energy ILStructure(i).Singlephase ILStructure(i).Threephase];
397         if (ILStructure(i).DeviceTech==1) & (A==B) & (Passive==1);           %In-line-Passive
398             DData(k)=ILStructure(i);
399             for l=1:length(DData)
400                 Devices(l)={DData(l).Name};
401                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
402             end
403             k=k+1;
404         elseif (ILStructure(i).DeviceTech==2) & (A==B) & (Mechanical==1);      %In-line-Mechanical
405             DData(k)=ILStructure(i);
406             for l=1:length(DData)
407                 Devices(l)={DData(l).Name};
408                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
409             end
410             k=k+1;
411         elseif (ILStructure(i).DeviceTech==3) & (A==B) & (Thyristor==1);      %In-line-Thyristor
412             DData(k)=ILStructure(i);
413             for l=1:length(DData)
414                 Devices(l)={DData(l).Name};
415                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
416             end
417             k=k+1;
418         elseif (ILStructure(i).DeviceTech==4) & (A==B) & (Transistor==1);     %In-line-Transistor
419             DData(k)=ILStructure(i);
420             for l=1:length(DData)
421                 Devices(l)={DData(l).Name};
422                 set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
423             end
424             k=k+1;
425         elseif isempty(Devices)
426             Devices={};
427             set(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'String',Devices);
428         end;
429     end;
430 else
431     Devices={};
432 end;
433
434 %SetDeviceNames(Devices)
435 value=get(findobj(findobj('Tag','DeviceGUI'),'Tag','DeviceLB'),'Value');
436
437 if isempty(Devices)
438     set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',num2str(V));
439     set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',num2str(P));
440     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',num2str(Q));
441     S=abs(P+j*Q);
442     set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
443     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String','no device');
444 else
445     set(findobj(findobj('Tag','DeviceGUI'),'Tag','DGRB'),'UserData',DData);
446     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Info'),'String',DData(value).Info);
447     set(findobj(findobj('Tag','DeviceGUI'),'Tag','V'),'String',DData(value).Voltage);
448     set(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String',DData(value).P);
449     set(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String',DData(value).Q);
450     P=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','P'),'String'));
451     Q=str2num(get(findobj(findobj('Tag','DeviceGUI'),'Tag','Q'),'String'));
452     S=abs(P+j*Q);
453     set(findobj(findobj('Tag','DeviceGUI'),'Tag','S'),'String',num2str(S));
454 end;
455
456 case 'Add_Device'
457     SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
458     ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
459     SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
460     ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
461
462     AddName=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddName'),'String');
463     AddP=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceP'),'String');
464     AddQ=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceQ'),'String');
465     AddV=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceV'),'String');
466     Energy=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','EnergyStorageCB'),'Value');
467     Singlephase=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','1phaseRB'),'Value');
468     Threephase=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','3phaseRB'),'Value');
469     DeviceConfig=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
470     DeviceInfo=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','InfoEdit'),'String');
471
472     Transistor=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','TransistorRB'),'Value');
473     Thyristor=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','ThyristorRB'),'Value');
474     Mechanical=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','MechanicalRB'),'Value');
475     Passive=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','PassiveRB'),'Value');
476     DG=get(findobj(findobj('Tag','AddDeviceGUI'),'Tag','DGRB'),'Value');
477
478     if Passive==1
479         DeviceTech=1;
480     elseif Mechanical==1
481         DeviceTech=2;
482     elseif Thyristor==1
483         DeviceTech=3;
484     elseif Transistor==1
485         DeviceTech=4;
486     else
487         DeviceTech=5;
488     end;
489
490     if isempty(AddV)
491         VF=0;

```



```

492     else
493         VF=1;
494     end;
495
496     if (isempty(AddP)==1) & (isempty(AddQ)==0)
497         AddS=j*str2num(AddQ);
498         PF=0;
499         QF=1;
500     elseif (isempty(AddP)==0) & (isempty(AddQ)==0)
501         AddS=str2num(AddP)+j*str2num(AddQ);
502         PF=1;
503         QF=1;
504     elseif (isempty(AddP)==0) & (isempty(AddQ)==1)
505         AddS=str2num(AddP);
506         PF=1;
507         QF=0;
508     else (isempty(AddP)==1) & (isempty(AddQ)==1)
509         PF=0;
510         QF=0;
511     end;
512
513     if (PF==0) & (QF==0)
514         set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceS'),'String','');
515     else
516         set(findobj(findobj('Tag','AddDeviceGUI'),'Tag','AddDeviceS'),'String',num2str(AddS));
517     end;
518
519     if (DeviceConfig==1)
520         sz=size(SeriesStructure);
521         SeriesStructure(sz(2)+1)=struct('DeviceTech','DeviceTech','Name',AddName,'Voltage',AddV,'VF','VF','P','P',AddP,
522         'PF','PF','Q','Q',AddQ,'QF','QF','Energy','Energy','Singlephase','Singlephase','Threephase','Threephase','Info','DeviceInfo');
523         set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData',SeriesStructure);
524     elseif (DeviceConfig==2)
525         sz=size(ShuntStructure);
526         ShuntStructure(sz(2)+1)=struct('DeviceTech','DeviceTech','Name',AddName,'Voltage',AddV,'VF','VF','P','P',AddP,
527         'PF','PF','Q','Q',AddQ,'QF','QF','Energy','Energy','Singlephase','Singlephase','Threephase','Threephase','Info','DeviceInfo');
528         set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData',ShuntStructure);
529     elseif (DeviceConfig==3)
530         sz=size(SSStructure);
531         SSStructure(sz(2)+1)=struct('DeviceTech','DeviceTech','Name',AddName,'Voltage',AddV,'VF','VF','P','P',AddP,'PF',
532         'PF','Q','Q',AddQ,'QF','QF','Energy','Energy','Singlephase','Singlephase','Threephase','Threephase','Info','DeviceInfo');
533         set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData',SSStructure);
534     elseif (DeviceConfig==4)
535         sz=size(ILStructure);
536         ILStructure(sz(2)+1)=struct('DeviceTech','DeviceTech','Name',AddName,'Voltage',AddV,'VF','VF','P','P',AddP,'PF',
537         'PF','Q','Q',AddQ,'QF','QF','Energy','Energy','Singlephase','Singlephase','Threephase','Threephase','Info','DeviceInfo');
538         set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData',ILStructure);
539     end;
540
541     fig = findobj('Tag','DeviceGUI');
542     saveas(fgure(fig),get(fig,'FileName'),'fig');
543
544     case 'GetStructure'
545         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value',1);
546         DeviceConfig=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
547         if DeviceConfig==1
548             SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
549             for l=1:length(SeriesStructure)
550                 Name(l)={SeriesStructure(l).Name};
551             end
552             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
553             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SeriesStructure);
554         elseif DeviceConfig==2
555             ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
556             for l=1:length(ShuntStructure)
557                 Name(l)={ShuntStructure(l).Name};
558             end
559             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
560             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ShuntStructure);
561         elseif DeviceConfig==3
562             SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
563             for l=1:length(SSStructure)
564                 Name(l)={SSStructure(l).Name};
565             end
566             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
567             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SSStructure);
568         elseif DeviceConfig==4
569             ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
570             for l=1:length(ILStructure)
571                 Name(l)={ILStructure(l).Name};
572             end
573             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
574             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ILStructure);
575         end;
576
577     case 'DisplayDevice'
578         value=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value');
579         devicestruct=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData');
580         VF=devicestruct(value).VF;
581         DeviceTech=devicestruct(value).DeviceTech;
582         if DeviceTech==1
583             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',1);
584             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
585             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
586             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
587             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0);
588         elseif DeviceTech==2
589             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',1);
590             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
591             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
592             set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);

```

```

593         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
594     elseif DeviceTech==3
595         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',1);
596         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
597         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
598         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
599         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
600     elseif DeviceTech==4
601         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',1);
602         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
603         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
604         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
605         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',0)
606     else
607         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value',1);
608         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value',0);
609         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value',0);
610         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value',0);
611         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value',0);
612     end;
613
614     if VF==1
615         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String',devicestruct(value).Voltage);
616     else
617         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String','');
618     end;
619     PF=devicestruct(value).PF;
620     QF=devicestruct(value).QF;
621
622     if (PF==1) & (QF==1)
623         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String',devicestruct(value).P);
624         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String',devicestruct(value).Q);
625         S=str2num(devicestruct(value).P)+j*str2num(devicestruct(value).Q);
626         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(S));
627     elseif (PF==1) & (QF==0)
628         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String',devicestruct(value).P);
629         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String','');
630         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',devicestruct(value).P);
631     elseif (PF==0) & (QF==1)
632         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String','');
633         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String',devicestruct(value).Q);
634         S=j*str2num(devicestruct(value).Q);
635         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(S));
636     else
637         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String','');
638         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String','');
639         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String','');
640     end;
641
642     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModName'),'String',devicestruct(value).Name);
643     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModInfo'),'String',devicestruct(value).Info);
644     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','EnergyStorageCB'),'Value',devicestruct(value).Energy);
645     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1 phaseRB'),'Value',devicestruct(value).Singlephase);
646     set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3 phaseRB'),'Value',devicestruct(value).Threephase);
647
648     case 'Modify'
649         value=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'Value');
650         devicestruct=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData');
651
652         ModName=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModName'),'String');
653         ModInfo=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModInfo'),'String');
654         ModP=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceP'),'String');
655         ModQ=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceQ'),'String');
656         ModV=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceV'),'String');
657
658         Energy=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','EnergyStorageCB'),'Value');
659         Singlephase=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','1 phaseRB'),'Value');
660         Threephase=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','3 phaseRB'),'Value');
661
662         Transistor=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','TransistorRB'),'Value');
663         Thyristor=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ThyristorRB'),'Value');
664         Mechanical=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','MechanicalRB'),'Value');
665         Passive=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','PassiveRB'),'Value');
666         DG=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DGRB'),'Value');
667
668         if Passive==1
669             DeviceTech=1;
670         elseif Mechanical==1
671             DeviceTech=2;
672         elseif Thyristor==1
673             DeviceTech=3;
674         elseif Transistor==1
675             DeviceTech=4;
676         else
677             DeviceTech=5;
678         end;
679
680         if isempty(ModV)
681             VF=0;
682         else
683             VF=1;
684         end;
685
686         if (isempty(ModP)==1) & (isempty(ModQ)==0)
687             ModS=j*str2num(ModQ);
688             PF=0;
689             QF=1;
690         elseif (isempty(ModP)==0) & (isempty(ModQ)==0)
691             ModS=str2num(ModP)+j*str2num(ModQ);
692             PF=1;
693             QF=1;

```



```

694     elseif (isempty(ModP)==0) & (isempty(ModQ)==1)
695         ModS=str2num(ModP);
696         PF=1;
697         QF=0;
698     else (isempty(ModP)==1) & (isempty(ModQ)==1)
699         PF=0;
700         QF=0;
701     end
702
703     if (PF==0) & (QF==0)
704         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String','');
705     else
706         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','ModDeviceS'),'String',num2str(ModS));
707     end;
708
709     DeviceConfig=get(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceConfigMenu'),'Value');
710     SeriesStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData');
711     ShuntStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData');
712     SSStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData');
713     ILStructure=get(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData');
714     if (DeviceConfig==1)
715         SeriesStructure(value)=struct('DeviceTech','DeviceTech','Name',ModName,'Voltage',ModV,'VF','P',ModP,
716         'PF','Q',ModQ,'QF','QF','Energy',Energy,'Singlephase','Singlephase','Threephase','Threephase','Info',ModInfo);
717         for l=1:length(SeriesStructure)
718             Name(l)={SeriesStructure(l).Name};
719         end
720         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
721         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SeriesStructure);
722         set(findobj(findobj('Tag','DeviceGUI'),'Tag','PassiveRB'),'UserData',SeriesStructure);
723     elseif (DeviceConfig==2)
724         ShuntStructure(value)=struct('DeviceTech','DeviceTech','Name',ModName,'Voltage',ModV,'VF','P',ModP,'PF'
725         'PF','Q',ModQ,'QF','QF','Energy',Energy,'Singlephase','Singlephase','Threephase','Threephase','Info',ModInfo);
726         for l=1:length(ShuntStructure)
727             Name(l)={ShuntStructure(l).Name};
728         end
729         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
730         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ShuntStructure);
731         set(findobj(findobj('Tag','DeviceGUI'),'Tag','MechanicalRB'),'UserData',ShuntStructure);
732     elseif (DeviceConfig==3)
733         SSStructure(value)=struct('DeviceTech','DeviceTech','Name',ModName,'Voltage',ModV,'VF','P',ModP,'PF',
734         'PF','Q',ModQ,'QF','QF','Energy',Energy,'Singlephase','Singlephase','Threephase','Threephase','Info',ModInfo);
735         for l=1:length(SSStructure)
736             Name(l)={SSStructure(l).Name};
737         end;
738         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
739         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',SSStructure);
740         set(findobj(findobj('Tag','DeviceGUI'),'Tag','ThyristorRB'),'UserData',SSStructure);
741     elseif (DeviceConfig==4)
742         ILStructure(value)=struct('DeviceTech','DeviceTech','Name',ModName,'Voltage',ModV,'VF','P',ModP,'PF',
743         'PF','Q',ModQ,'QF','QF','Energy',Energy,'Singlephase','Singlephase','Threephase','Threephase','Info',ModInfo);
744         for l=1:length(ILStructure)
745             Name(l)={ILStructure(l).Name};
746         end
747         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'String',Name);
748         set(findobj(findobj('Tag','ModDeviceGUI'),'Tag','DeviceLB'),'UserData',ILStructure);
749         set(findobj(findobj('Tag','DeviceGUI'),'Tag','TransistorRB'),'UserData',ILStructure);
750     end;
751     fig = findobj('Tag','DeviceGUI');
752     saveas(fgure(fig),get(fig,'FileName'),'fig');
753
754     case 'OpenAddDevice'
755         fig=findobj('Tag','AddDeviceGUI');
756         if isempty(fig)==1
757             open('AddDevice.fig');
758         end;
759     case 'OpenModifyDevice'
760         fig=findobj('Tag','ModDeviceGUI');
761         if isempty(fig)==1
762             open('ModifyDevice.fig');
763         end;
764
765     case 'CloseAddDevice'
766         fig = findobj('Tag','AddDeviceGUI');
767         close(fgure(fig));
768
769     case 'CloseModDevice'
770         fig = findobj('Tag','ModDeviceGUI');
771         close(fgure(fig));
772
773     case 'CloseDeviceLib'
774         fig = findobj('Tag','DeviceGUI');
775         saveas(fgure(fig),get(fig,'FileName'),'fig');
776         close(fgure(fig));
777
778     end;

```

## B.4 TESAT CODE

### B.4.1 LFAC.m

```

1 function lfac(action)
2 %Main Loadflow for a single three-phase ac line GUI
3
4 switch(action)
5
6 case 'VoltageRegulation'
7     set(gcbo,'Value',1);
8     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','%');
9     set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
10    set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
11    set(findobj(findobj('Tag','lfacGUI'),'Tag','ATlrb'),'Value',0);
12    set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
13    set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','0');
14 case 'Loss'
15     set(gcbo,'Value',1);
16     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','kW');
17     set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
18     set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
19     set(findobj(findobj('Tag','lfacGUI'),'Tag','ATlrb'),'Value',0);
20     set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
21     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','10');
22
23 case 'CCL'
24     set(gcbo,'Value',1);
25     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','R');
26     set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
27     set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
28     set(findobj(findobj('Tag','lfacGUI'),'Tag','ATlrb'),'Value',0);
29     set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
30     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','100000');
31
32 case 'ATT'
33     set(gcbo,'Value',1);
34     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcontunit'),'String','');
35     set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value',0);
36     set(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value',0);
37     set(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value',0);
38     set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
39     set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','1');
40
41 case 'Posslider'
42     length=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','lengthe'),'String'));
43     UserData=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData');
44     set(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Max',length);
45     set(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Sliderstep',[1/length 10/length]);
46     set(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Min',1);
47     position=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Value'));
48     if position>length
49         set(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Value',1);
50         set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String','1');
51         UserData(38)=1;
52     else
53         set(findobj(findobj('Tag','lfacGUI'),'Tag','Disppos'),'String',num2str(position));
54         UserData(38)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Value'));
55     end;
56     set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData',UserData);
57     Change2dPlots([1 0 0])
58
59 case 'Yearslider'
60     years=str2num(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearse'),'String'));
61     UserData=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData');
62     set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Max',years);
63     set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Sliderstep',[1/years 10/years]);
64     set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Min',0);
65     year=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Value'));
66     if year>years
67         set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Value',0);
68         set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
69         UserData(39)=0;
70     else
71         set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String',num2str(year));
72         UserData(39)=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Value'));
73     end;
74     set(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'UserData',UserData);
75     Change2dPlots([0 1 0])
76
77 case 'Contourslider'
78     vr=get(findobj(findobj('Tag','lfacGUI'),'Tag','vrrb'),'Value');
79     loss=get(findobj(findobj('Tag','lfacGUI'),'Tag','lossrb'),'Value');
80     ccl=get(findobj(findobj('Tag','lfacGUI'),'Tag','cclrb'),'Value');
81     ati=get(findobj(findobj('Tag','lfacGUI'),'Tag','ATlrb'),'Value');
82     if vr==1
83         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
84         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.005 0.05]);
85         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',-100);
86         valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
87         if valuecontslider>100
88             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
89         end;
90         set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
91     elseif loss==1

```



```

92         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
93         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
94         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
95         valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
96         if valuecontslider==0
97             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
98         else
99             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
100         end;
101     elseif ccl==1
102         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
103         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
104         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
105         valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
106         if valuecontslider==0
107             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
108         else;
109             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*10000));
110         end;
111     elseif ati==1
112         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10);
113         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
114         set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',1);
115         valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
116         if valuecontslider==0
117             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
118         elseif valuecontslider>10
119             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
120         else;
121             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
122         end;
123     end;
124
125 case 'r1'
126     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
127                                     % erase input on error
128 case 'r2'
129     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
130                                     % erase input on error
131 case 'r3'
132     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
133                                     % erase input on error
134 case 'x1'
135     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
136                                     % erase input on error
137 case 'x2'
138     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
139                                     % erase input on error
140 case 'x3'
141     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
142                                     % erase input on error
143 case 'b1'
144     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
145                                     % erase input on error
146 case 'b2'
147     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
148                                     % erase input on error
149 case 'b3'
150     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
151                                     % erase input on error
152 case 'cost1'
153     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
154                                     % erase input on error
155 case 'cost2'
156     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
157                                     % erase input on error
158 case 'cost3'
159     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
160                                     % erase input on error
161 case 'sysV'
162     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
163                                     % erase input on error
164 case 'Vs'
165     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
166                                     % erase input on error
167 case 'length'
168     oldvalue=get(findobj(findobj('Tag','lfacGUI'),'Tag','possider'),'Max');
169     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
170                                     % erase input on error
171     newvalue=str2num(get(gcbo,'String'));
172     if newvalue<=1
173         set(gcbo,'String',oldvalue);
174     end;
175 case 'LG'
176     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
177                                     % erase input on error
178 case 'IL'
179     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
180                                     % erase input on error
181 case 'years'
182     oldvalue=get(findobj(findobj('Tag','lfacGUI'),'Tag','yearsider'),'Max');
183     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
184                                     % erase input on error
185     newvalue=str2num(get(gcbo,'String'));
186     if newvalue<=0
187         set(gcbo,'String',oldvalue);
188     elseif newvalue>25
189         set(gcbo,'String',oldvalue);
190     end;
191 case 'stationcost'
192     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
193                                     % erase input on error
194 case 'interest'
195     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
196                                     % erase input on error
197 case 'ALF'
198     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
199                                     % erase input on error
200 case 'ampc1'
201     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
202                                     % erase input on error
203 case 'ampc2'
204     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
205                                     % erase input on error
206 case 'ampc3'
207     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
208                                     % erase input on error
209 case 'w1'
210     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
211                                     % erase input on error
212     value=str2num(get(gcbo,'String'));
213     if value>1
214         set(gcbo,'String','');
215     end;
216 case 'w2'
217     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
218                                     % erase input on error

```

```

193     value=str2num(get(gcbo,'String'));
194     if value>1
195         set(gcbo,'String','');
196     end;
197 case 'w3'
198     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
199     value=str2num(get(gcbo,'String'));
200     if value>1
201         set(gcbo,'String','');
202     end;
203 case 'w4'
204     set(gcbo,'String',num2str(str2num(get(gcbo,'String'))))
205     value=str2num(get(gcbo,'String'));
206     if value>1
207         set(gcbo,'String','');
208     end;
209
210 case 'Run'
211     %Initialize variables
212     clear all;
213     plotc1cont=[];
214     plotc2cont=[];
215     plotc3cont=[];
216
217     GC=str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','gclb'),'String'));
218     namec1=get(findobj(findobj('Tag','IfacGUI'),'Tag','c1e'),'String');
219     namec2=get(findobj(findobj('Tag','IfacGUI'),'Tag','c2e'),'String');
220     namec3=get(findobj(findobj('Tag','IfacGUI'),'Tag','c3e'),'String');
221
222     UserData(1)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','r1e'),'String')));
223     UserData(2)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','r2e'),'String')));
224     UserData(3)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','r3e'),'String')));
225     UserData(4)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','x1e'),'String')));
226     UserData(5)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','x2e'),'String')));
227     UserData(6)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','x3e'),'String')));
228     UserData(7)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','b1e'),'String')))*1e-6;
229     UserData(8)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','b2e'),'String')))*1e-6;
230     UserData(9)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','b3e'),'String')))*1e-6;
231     UserData(10)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','costc1e'),'String')));
232     UserData(11)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','costc2e'),'String')));
233     UserData(12)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','costc3e'),'String')));
234     UserData(13)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','ampc1e'),'String')));
235     UserData(14)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','ampc2e'),'String')));
236     UserData(15)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','ampc3e'),'String')));
237     UserData(16)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','lengthe'),'String')));
238     UserData(17)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','yarsee'),'String')));
239     UserData(18)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','ile'),'String')))*1e6;
240     UserData(19)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','lge'),'String')))/100;
241     UserData(20)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','pfe'),'String')));
242     UserData(22)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','sysve'),'String')))*1e3;
243     UserData(21)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','sendve'),'String'))/100)*UserData(22);
244     UserData(23)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','inte'),'String')))/100;
245     UserData(24)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','alfe'),'String')))^2;
246     UserData(25)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','ccse'),'String')));
247     UserData(26)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','w1e'),'String')));
248     UserData(27)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','w2e'),'String')));
249     UserData(28)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','w3e'),'String')));
250     UserData(29)=(str2num(get(findobj(findobj('Tag','IfacGUI'),'Tag','w4e'),'String')));
251     UserData(30)=get(findobj(findobj('Tag','IfacGUI'),'Tag','vrrb'),'Value');
252     UserData(31)=get(findobj(findobj('Tag','IfacGUI'),'Tag','lossrb'),'Value');
253     UserData(32)=get(findobj(findobj('Tag','IfacGUI'),'Tag','cc1rb'),'Value');
254     UserData(33)=get(findobj(findobj('Tag','IfacGUI'),'Tag','AT1rb'),'Value');
255
256     UserData(35)=get(findobj(findobj('Tag','IfacGUI'),'Tag','CCC1cb'),'Value');
257     UserData(36)=get(findobj(findobj('Tag','IfacGUI'),'Tag','CCC2cb'),'Value');
258     UserData(37)=get(findobj(findobj('Tag','IfacGUI'),'Tag','CCC3cb'),'Value');
259     SetSliders(UserData);
260
261     UserData(34)=round(get(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value'));
262     UserData(38)=round(get(findobj(findobj('Tag','IfacGUI'),'Tag','poss1ider'),'Value'));
263     UserData(39)=round(get(findobj(findobj('Tag','IfacGUI'),'Tag','years1ider'),'Value'));
264
265     %ac Voltage regulation/loss/CCL/ATI functions
266     [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
267         =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);
268
269     %3d plots of voltage regulation/loss/CCL/ATI
270     plot3dac(XData123,YData,namec1,namec2,namec3,UserData,Data)
271
272     %2d plots of cross-section through 3d voltage regulation/loss/CCL/ATI plots
273     plot2dac(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)
274
275     set(findobj(findobj('Tag','IfacGUI'),'Tag','vrrb'),'UserData',UserData);
276     set(findobj(findobj('Tag','IfacGUI'),'Tag','lossrb'),'UserData',Data);
277     set(findobj(findobj('Tag','IfacGUI'),'Tag','cc1rb'),'UserData',XData123);
278     set(findobj(findobj('Tag','IfacGUI'),'Tag','AT1rb'),'UserData',YData);
279     set(findobj(findobj('Tag','IfacGUI'),'Tag','rpb'),'UserData',XD);
280     set(findobj(findobj('Tag','IfacGUI'),'Tag','resetpb'),'UserData',XData);
281     set(findobj(findobj('Tag','IfacGUI'),'Tag','closepb'),'UserData',LR);
282     set(findobj(findobj('Tag','IfacGUI'),'Tag','c1e'),'UserData',plotc1cont);
283     set(findobj(findobj('Tag','IfacGUI'),'Tag','c2e'),'UserData',plotc2cont);
284     set(findobj(findobj('Tag','IfacGUI'),'Tag','c3e'),'UserData',plotc3cont);
285
286 case 'reset'
287     set(findobj(findobj('Tag','IfacGUI'),'Tag','vrrb'),'Value',1);
288     set(findobj(findobj('Tag','IfacGUI'),'Tag','lossrb'),'Value',0);
289     set(findobj(findobj('Tag','IfacGUI'),'Tag','cc1rb'),'Value',0);
290     set(findobj(findobj('Tag','IfacGUI'),'Tag','AT1rb'),'Value',0);
291
292     set(findobj(findobj('Tag','IfacGUI'),'Tag','poss1ider'),'Value',1);
293     set(findobj(findobj('Tag','IfacGUI'),'Tag','poss1ider'),'Max',10);

```



```

294 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispos'),'String','1');
295 set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsldier'),'Value',0);
296 set(findobj(findobj('Tag','lfacGUI'),'Tag','yearsldier'),'Max',25);
297 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
298 set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',1);
299 set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
300 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcnt'),'String','1');
301
302 set(findobj(findobj('Tag','lfacGUI'),'Tag','c1e'),'String','');
303 set(findobj(findobj('Tag','lfacGUI'),'Tag','r1e'),'String','');
304 set(findobj(findobj('Tag','lfacGUI'),'Tag','x1e'),'String','');
305 set(findobj(findobj('Tag','lfacGUI'),'Tag','b1e'),'String','');
306 set(findobj(findobj('Tag','lfacGUI'),'Tag','costc1e'),'String','');
307 set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc1e'),'String','');
308 set(findobj(findobj('Tag','lfacGUI'),'Tag','c1t'),'BackgroundColor',[ 8314 8157 7843]);
309 set(findobj(findobj('Tag','lfacGUI'),'Tag','c2e'),'String','');
310 set(findobj(findobj('Tag','lfacGUI'),'Tag','r2e'),'String','');
311 set(findobj(findobj('Tag','lfacGUI'),'Tag','x2e'),'String','');
312 set(findobj(findobj('Tag','lfacGUI'),'Tag','b2e'),'String','');
313 set(findobj(findobj('Tag','lfacGUI'),'Tag','costc2e'),'String','');
314 set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc2e'),'String','');
315 set(findobj(findobj('Tag','lfacGUI'),'Tag','c2t'),'BackgroundColor',[ 8314 8157 7843]);
316 set(findobj(findobj('Tag','lfacGUI'),'Tag','c3e'),'String','');
317 set(findobj(findobj('Tag','lfacGUI'),'Tag','r3e'),'String','');
318 set(findobj(findobj('Tag','lfacGUI'),'Tag','x3e'),'String','');
319 set(findobj(findobj('Tag','lfacGUI'),'Tag','b3e'),'String','');
320 set(findobj(findobj('Tag','lfacGUI'),'Tag','costc3e'),'String','');
321 set(findobj(findobj('Tag','lfacGUI'),'Tag','ampc3e'),'String','');
322 set(findobj(findobj('Tag','lfacGUI'),'Tag','c3t'),'BackgroundColor',[ 8314 8157 7843]);
323 set(findobj(findobj('Tag','lfacGUI'),'Tag','sysve'),'String','');
324 set(findobj(findobj('Tag','lfacGUI'),'Tag','sendve'),'String','');
325 set(findobj(findobj('Tag','lfacGUI'),'Tag','lengthe'),'String','');
326 set(findobj(findobj('Tag','lfacGUI'),'Tag','lge'),'String','');
327 set(findobj(findobj('Tag','lfacGUI'),'Tag','ile'),'String','');
328 set(findobj(findobj('Tag','lfacGUI'),'Tag','yearse'),'String','');
329 set(findobj(findobj('Tag','lfacGUI'),'Tag','inte'),'String','');
330 set(findobj(findobj('Tag','lfacGUI'),'Tag','gclb'),'Value',1);
331 set(findobj(findobj('Tag','lfacGUI'),'Tag','alfe'),'String','');
332 set(findobj(findobj('Tag','lfacGUI'),'Tag','ccse'),'String','');
333 set(findobj(findobj('Tag','lfacGUI'),'Tag','w1e'),'String','');
334 set(findobj(findobj('Tag','lfacGUI'),'Tag','w2e'),'String','');
335 set(findobj(findobj('Tag','lfacGUI'),'Tag','w3e'),'String','');
336 set(findobj(findobj('Tag','lfacGUI'),'Tag','w4e'),'String','');
337 set(findobj(findobj('Tag','lfacGUI'),'Tag','pfe'),'String','');
338
339 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl1axes'))
340 cla reset
341 set(gca,'Tag','vrl1axes');
342 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl2axes'))
343 cla reset
344 set(gca,'Tag','vrl2axes');
345 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrl3axes'))
346 cla reset
347 set(gca,'Tag','vrl3axes');
348 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlposaxes'))
349 cla reset
350 set(gca,'Tag','vrlposaxes');
351 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlyearaxes'))
352 cla reset
353 set(gca,'Tag','vrlyearaxes');
354 axes(findobj(findobj('Tag','lfacGUI'),'Tag','vrlcontaxes'))
355 cla reset
356 set(gca,'Tag','vrlcontaxes');
357
358 if get(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value')
359     set(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value',0)
360     fig = findobj('Tag','3dvrlc1fig');
361     if ~isempty(fig)
362         close(fgure(fig));
363     end;
364     fig = findobj('Tag','3dvrlc2fig');
365     if ~isempty(fig)
366         close(fgure(fig));
367     end;
368     fig = findobj('Tag','3dvrlc3fig');
369     if ~isempty(fig)
370         close(fgure(fig));
371     end;
372     fig = findobj('Tag','2dvrlpointloadfig');
373     if ~isempty(fig)
374         close(fgure(fig));
375     end;
376     fig = findobj('Tag','2dvrlposfig');
377     if ~isempty(fig)
378         close(fgure(fig));
379     end;
380     fig = findobj('Tag','2dvrlcontfig');
381     if ~isempty(fig)
382         close(fgure(fig));
383     end;
384 end;
385
386 case 'CloseHighResolutionFigures'
387     fig = findobj('Tag','3dvrlc1fig');
388     if ~isempty(fig)
389         close(fgure(fig));
390     end;
391     fig = findobj('Tag','3dvrlc2fig');
392     if ~isempty(fig)
393         close(fgure(fig));
394     end;

```

```

395     fig = findobj('Tag','3dvrlc3fig');
396     if ~isempty(fig)
397         close.figure(fig);
398     end;
399     fig = findobj('Tag','2dvrlpointloadfig');
400     if ~isempty(fig)
401         close.figure(fig);
402     end;
403     fig = findobj('Tag','2dvrlposfig');
404     if ~isempty(fig)
405         close.figure(fig);
406     end;
407     fig = findobj('Tag','2dvrlcontfig');
408     if ~isempty(fig)
409         close.figure(fig);
410     end;
411
412     case 'close'
413     if get(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value')
414         set(findobj(findobj('Tag','lfacGUI'),'Tag','hrfcb'),'Value',0)
415         fig = findobj('Tag','3dvrlc1fig');
416         if ~isempty(fig)
417             close.figure(fig);
418         end;
419         fig = findobj('Tag','3dvrlc2fig');
420         if ~isempty(fig)
421             close.figure(fig);
422         end;
423         fig = findobj('Tag','3dvrlc3fig');
424         if ~isempty(fig)
425             close.figure(fig);
426         end;
427         fig = findobj('Tag','2dvrlpointloadfig');
428         if ~isempty(fig)
429             close.figure(fig);
430         end;
431         fig = findobj('Tag','2dvrlposfig');
432         if ~isempty(fig)
433             close.figure(fig);
434         end;
435         fig = findobj('Tag','2dvrlcontfig');
436         if ~isempty(fig)
437             close.figure(fig);
438         end;
439     end;
440     fig = findobj('Tag','lfacGUI');
441     saveas.figure(fig, get(fig, 'FileName'), 'fig')
442     close.figure(fig);
443
444     function SetSliders(UserData)
445         length=UserData(16);
446         years=UserData(17);
447         vr=UserData(30);
448         loss=UserData(31);
449         ccl=UserData(32);
450         ati=UserData(33);
451
452         set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Max',length);
453         set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
454         set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Min',1);
455         position=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value'));
456         if position>length
457             set(findobj(findobj('Tag','lfacGUI'),'Tag','posslider'),'Value',1);
458             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispos'),'String','1');
459         else
460             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispos'),'String',num2str(position));
461         end;
462
463         set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Max',years);
464         set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
465         set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Min',0);
466         valueyearslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value'));
467         if valueyearslider>years
468             set(findobj(findobj('Tag','lfacGUI'),'Tag','yearslider'),'Value',0);
469             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String','0');
470         else
471             set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispyear'),'String',num2str(valueyearslider));
472         end;
473
474         if vr==1
475             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',100);
476             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
477             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',-100);
478             valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
479             if valuecontslider>100
480                 set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',0);
481                 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','0');
482             else
483                 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
484             end;
485         elseif loss==1
486             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Max',10e3);
487             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
488             set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Min',0);
489             valuecontslider=round(get(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value'));
490             if valuecontslider==0
491                 set(findobj(findobj('Tag','lfacGUI'),'Tag','contslider'),'Value',10);
492                 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String','10');
493             else
494                 set(findobj(findobj('Tag','lfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
495             end;

```



```

496     elseif ccl==1
497         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Max',10e3);
498         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
499         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Min',0);
500         valuecontslider=round(get(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value'));
501         if valuecontslider==0
502             set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value',10);
503             set(findobj(findobj('Tag','IfacGUI'),'Tag','Dispcont'),'String','100000');
504         else
505             set(findobj(findobj('Tag','IfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*100000));
506         end;
507     elseif ati==1
508         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Max',10);
509         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
510         set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Min',1);
511         valuecontslider=round(get(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value'));
512         if valuecontslider==0
513             set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value',1);
514         elseif valuecontslider>10
515             set(findobj(findobj('Tag','IfacGUI'),'Tag','contslider'),'Value',1);
516         else;
517             set(findobj(findobj('Tag','IfacGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
518         end;
519     end;
520
521 function [Data,XData,XData123,L,R,XD,YData,plotc1cont,plotc2cont,plotc3cont]
522             =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);
523
524     rc1=UserData(1);
525     rc2=UserData(2);
526     rc3=UserData(3);
527     xc1=UserData(4);
528     xc2=UserData(5);
529     xc3=UserData(6);
530     bc1=UserData(7);
531     bc2=UserData(8);
532     bc3=UserData(9);
533     costc1=UserData(10);
534     costc2=UserData(11);
535     costc3=UserData(12);
536     ampc1=UserData(13);
537     ampc2=UserData(14);
538     ampc3=UserData(15);
539     length=UserData(16);
540     years=UserData(17);
541     IL=UserData(18);
542     LG=UserData(19);
543     pf=UserData(20);
544     Vs=UserData(21);
545     sysV=UserData(22);
546     Interest=UserData(23);
547     ELF=UserData(24);
548     CCS=UserData(25);
549     w1=UserData(26);
550     w2=UserData(27);
551     w3=UserData(28);
552     w4=UserData(29);
553     vr=UserData(30);
554     loss=UserData(31);
555     ccl=UserData(32);
556     ati=UserData(33);
557     contour=UserData(34);
558     comp1=UserData(35);
559     comp2=UserData(36);
560     comp3=UserData(37);
561
562     Sbase=100e6;
563     Zbase=(sysV^2)/Sbase;
564     Vspu=Vs/sysV;
565
566     for c=1:(years+1)
567         XData(c)=(IL/1e6)*(1+LG)^(c-1);
568         DF(c)=(1+Interest)^(c-1);
569         CLA(c)=8760*(GC(c)/100)*ELF;
570     end;
571
572     XD1=zeros([length,1]);
573     XD2=zeros([length,1]);
574     XD3=zeros([length,1]);
575
576     testLR1=zeros([years+1,1]);
577     testLR2=zeros([years+1,1]);
578     testLR3=zeros([years+1,1]);
579
580     contourmax1=0;
581     contourmax2=0;
582     contourmax3=0;
583     contourmin1=10;
584     contourmin2=10;
585     contourmin3=10;
586
587     SILlower=0;
588     SILupper=0;
589
590     yc1=bc1*j;
591     Zc1=sqrt((rc1+xc1*j)/yc1);
592     lamda1=sqrt((rc1+xc1*j)*yc1);
593
594     yc2=bc2*j;
595     Zc2=sqrt((rc2+xc2*j)/yc2);
596     lamda2=sqrt((rc2+xc2*j)*yc2);

```

%Power base  
%Calculate Zbase  
%Calculate the sending voltage in p.u.  
  
%Set up XData vector for 3d plot  
%Discount Factor  
%Cost of losses/kW/Annum [R]

```

597 yc3=bc3*j;
598 Zc3=sqrt((rc3+xc3*j)/yc3);
599 lamda3=sqrt((rc3+xc3*j)*yc3);
600
601 YData=1:length;
602 for j=1:(years+1)
603     LR1(j)=length;
604     LR2(j)=length;
605     LR3(j)=length;
606 end;
607
608 h = waitbar(0,'Calculating matrices...');
609
610 for i=1:length
611     waitbar(i/length);
612
613     Z1=Zc1*sinh(lamda1*i);
614     Y1=(2*tanh(lamda1*i/2))/Zc1;
615     R1pu=real(Z1)/Zbase;
616     X1pu=imag(Z1)/Zbase;
617     B1pu=imag(Y1)*Zbase;
618
619     Z2=Zc2*sinh(lamda2*i);
620     Y2=(2*tanh(lamda2*i/2))/Zc2;
621     R2pu=real(Z2)/Zbase;
622     X2pu=imag(Z2)/Zbase;
623     B2pu=imag(Y2)*Zbase;
624
625     Z3=Zc3*sinh(lamda3*i);
626     Y3=(2*tanh(lamda3*i/2))/Zc3;
627     R3pu=real(Z3)/Zbase;
628     X3pu=imag(Z3)/Zbase;
629     B3pu=imag(Y3)*Zbase;
630
631     Z1sqrpu=R1pu^2+X1pu^2;
632     Z2sqrpu=R2pu^2+X2pu^2;
633     Z3sqrpu=R3pu^2+X3pu^2;
634
635     K1=1-B1pu*X1pu+(B1pu^2*Z1sqrpu)/4;
636     K2=1-B2pu*X2pu+(B2pu^2*Z2sqrpu)/4;
637     K3=1-B3pu*X3pu+(B3pu^2*Z3sqrpu)/4;
638
639     XData1(i,:)=XData;
640     XData2(i,:)=XData;
641     XData3(i,:)=XData;
642
643     for j=1:(years+1)
644         Srpu=(IL/Sbase)*(1+LG)^(j-1);
645         Prpu=((IL*pf)/Sbase)*(1+LG)^(j-1);
646         Qrpu=sqrt(Srpu^2-Prpu^2);
647
648         F1=(Vspu^2)-2*R1pu*Prpu-Qrpu*(2*X1pu-B1pu*Z1sqrpu);
649         F2=(Vspu^2)-2*R2pu*Prpu-Qrpu*(2*X2pu-B2pu*Z2sqrpu);
650         F3=(Vspu^2)-2*R3pu*Prpu-Qrpu*(2*X3pu-B3pu*Z3sqrpu);
651
652         Vr1pu(i,j)=sqrt((F1+sqrt(F1^2-4*Z1sqrpu*(Prpu^2+Qrpu^2)*K1))/(2*K1));
653         Vr2pu(i,j)=sqrt((F2+sqrt(F2^2-4*Z2sqrpu*(Prpu^2+Qrpu^2)*K2))/(2*K2));
654         Vr3pu(i,j)=sqrt((F3+sqrt(F3^2-4*Z3sqrpu*(Prpu^2+Qrpu^2)*K3))/(2*K3));
655
656         if isreal(Vr1pu(i,j))
657             Vreg1(i,j)=(Vspu-abs(Vr1pu(i,j)))*100;
658             Ploss1(i,j)=((R1pu/Z1sqrpu)*(Vspu^2-Vr1pu(i,j)^2-
659                 2*(R1pu*Prpu+X1pu*Qrpu)+B1pu*X1pu*Vr1pu(i,j)^2))*1E5;
660             ECL1(i,j)=Ploss1(i,j)*CLA(j);
661             PVCL1(i,j)=ECL1(i,j)/DF(j);
662             PVCL1(:,1)=0;
663             jrepeat1=j;
664         else
665             if testLR1(j)==0
666                 LR1(j)=i-1;
667                 testLR1(j)=1;
668             end;
669             Vreg1(i,j)=Vreg1(i,jrepeat1);
670             Ploss1(i,j)=Ploss1(i,jrepeat1);
671             PVCL1(i,j)=0;
672             XData1(i,j)=XData(jrepeat1);
673             XD1(i)=1;
674         end;
675
676         if isreal(Vr2pu(i,j))
677             Vreg2(i,j)=(Vspu-abs(Vr2pu(i,j)))*100;
678             Ploss2(i,j)=((R2pu/Z2sqrpu)*(Vspu^2-Vr2pu(i,j)^2-
679                 2*(R2pu*Prpu+X2pu*Qrpu)+B2pu*X2pu*Vr2pu(i,j)^2))*1E5;
680             ECL2(i,j)=Ploss2(i,j)*CLA(j);
681             PVCL2(i,j)=ECL2(i,j)/DF(j);
682             PVCL2(:,1)=0;
683             jrepeat2=j;
684         else
685             if testLR2(j)==0
686                 LR2(j)=i-1;
687                 testLR2(j)=1;
688             end;
689             Vreg2(i,j)=Vreg2(i,jrepeat2);
690             Ploss2(i,j)=Ploss2(i,jrepeat2);
691             PVCL2(i,j)=0;
692             XData2(i,j)=XData(jrepeat2);
693             XD2(i)=1;
694         end;
695
696         if isreal(Vr3pu(i,j))
697             Vreg3(i,j)=(Vspu-abs(Vr3pu(i,j)))*100;

```



```

698     Ploss3(i,j)=(R3 pu/Z3sqpu)*(Vspu^2-Vr3 pu(i,j)^2-
699         2*(R3 pu*Prpu+X3 pu*Qrpu)+B3 pu*X3 pu*Vr3 pu(i,j)^2))*1E5;
700     ECL3(i,j)=Ploss3(i,j)*CLA(j);
701     PVCL3(i,j)=ECL3(i,j)/DF(j);
702     PVCL3(:,1)=0;
703     jrepeat3=j;
704 else
705     if testLR3(j)==0
706         LR3(j)=i-1;
707         testLR3(j)=1;
708     end;
709     Vreg3(i,j)=Vreg3(i,jrepeat3);
710     Ploss3(i,j)=Ploss3(i,jrepeat3);
711     PVCL3(i,j)=0;
712     XData3(i,j)=XData(jrepeat3);
713     XD3(i)=1;
714 end;
715
716 if (vr==1) & (contour>0) & (Vreg1(i,j)>=contour)
717     plotc1cont=1;
718 elseif (vr==1) & (contour<=0) & (Vreg1(i,j)<=contour)
719     plotc1cont=1;
720 elseif (loss==1) & (Ploss1(i,j)>=contour)
721     plotc1cont=1;
722 end;
723
724 if (vr==1) & (contour>0) & (Vreg2(i,j)>=contour)
725     plotc2cont=1;
726 elseif (vr==1) & (contour<=0) & (Vreg2(i,j)<=contour)
727     plotc2cont=1;
728 elseif (loss==1) & (Ploss2(i,j)>=contour)
729     plotc2cont=1;
730 end;
731
732 if (vr==1) & (contour>0) & (Vreg3(i,j)>=contour)
733     plotc3cont=1;
734 elseif (vr==1) & (contour<=0) & (Vreg3(i,j)<=contour)
735     plotc3cont=1;
736 elseif (loss==1) & (Ploss3(i,j)>=contour)
737     plotc3cont=1;
738 end;
739
740 Capacityc1(i,j)=(costc1*i)/(sqrt(3)*sysV*1e-3*ampc1*1*i);
741 Capacityc2(i,j)=(costc2*i)/(sqrt(3)*sysV*1e-3*ampc2*1*i);
742 Capacityc3(i,j)=(costc3*i)/(sqrt(3)*sysV*1e-3*ampc3*1*i);
743
744 SILc1(i,j)=(costc1*sqrt(abs(Z1)/imag(Y1)))/((sysV/1e3)^2);
745 SILc2(i,j)=(costc2*sqrt(abs(Z2)/imag(Y2)))/((sysV/1e3)^2);
746 SILc3(i,j)=(costc3*sqrt(abs(Z3)/imag(Y3)))/((sysV/1e3)^2);
747
748 end;
749
750 CPVCL1(i,:)=cumsum(PVCL1(i,:));
751 CPVCL2(i,:)=cumsum(PVCL2(i,:));
752 CPVCL3(i,:)=cumsum(PVCL3(i,:));
753
754 if comp1==1
755     PVCCL1(i,:)=(CPVCL1(i,:)+costc1*i+CCS)/1e6;
756 else
757     PVCCL1(i,:)=(CPVCL1(i,:)+costc1*i)/1e6;
758 end;
759 if comp2==1
760     PVCCL2(i,:)=(CPVCL2(i,:)+costc2*i+CCS)/1e6;
761 else
762     PVCCL2(i,:)=(CPVCL2(i,:)+costc2*i)/1e6;
763 end;
764 if comp3==1
765     PVCCL3(i,:)=(CPVCL3(i,:)+costc3*i+CCS)/1e6;
766 else
767     PVCCL3(i,:)=(CPVCL3(i,:)+costc3*i)/1e6;
768 end;
769
770 if SILc1(i,:)>SILlower
771     SILlower=SILc1(i,1);
772 elseif SILc2(i,:)>SILlower
773     SILlower=SILc2(i,1);
774 elseif SILc3(i,:)>SILlower
775     SILlower=SILc3(i,1);
776 end;
777 if SILc1(i,:)<SILc2(i,:) & SILc1(i,:)<SILc3(i,:)
778     SILupper=SILc1(i,1);
779 elseif SILc2(i,:)<SILc3(i,:)
780     SILupper=SILc2(i,1);
781 else
782     SILupper=SILc3(i,1);
783 end;
784
785 for n=1:(years+1)
786     if (ccl==1) & (PVCCL1(i,n)>=contour*10000) & (contour*10000>PVCCL1(1,1))
787         plotc1cont=1;
788     end;
789     if (ccl==1) & (PVCCL2(i,n)>=contour*10000) & (contour*10000>PVCCL2(1,1))
790         plotc2cont=1;
791     end;
792     if (ccl==1) & (PVCCL3(i,n)>=contour*10000) & (contour*10000>PVCCL3(1,1))
793         plotc3cont=1;
794     end;
795 end;
796
797 end;
798

```

%Capacity Matrix of conductor 1  
%Capacity Matrix of conductor 2  
%Capacity Matrix of conductor 3

%Cumulative Present Value Cost of Losses for Conductor 1  
%Cumulative Present Value Cost of Losses for Conductor 2  
%Cumulative Present Value Cost of Losses for Conductor 3

```

799 close(h);
800 if isreal(Vr1pu)
801     set(findobj(findobj('Tag','IfacGUI'),'Tag','c1t'),'BackgroundColor',[ 8314 .8157 .7843]);
802 else
803     set(findobj(findobj('Tag','IfacGUI'),'Tag','c1t'),'BackgroundColor','r');
804 end;
805 if isreal(Vr2pu)
806     set(findobj(findobj('Tag','IfacGUI'),'Tag','c2t'),'BackgroundColor',[ 8314 .8157 .7843]);
807 else
808     set(findobj(findobj('Tag','IfacGUI'),'Tag','c2t'),'BackgroundColor','r');
809 end;
810 if isreal(Vr3pu)
811     set(findobj(findobj('Tag','IfacGUI'),'Tag','c3t'),'BackgroundColor',[ 8314 .8157 .7843]);
812 else
813     set(findobj(findobj('Tag','IfacGUI'),'Tag','c3t'),'BackgroundColor','r');
814 end;
815
816 %Calculate upper, median and lower values for LCC KPI
817 LCClower=0;
818 LCCupper=0;
819 if PVCCL1(length,years+1)>PVCCL2(length,years+1) & PVCCL1(length,years+1)>PVCCL3(length,years+1)
820     LCClower=PVCCL1(length,years+1)+1;
821 elseif PVCCL2(length,years+1)>PVCCL3(length,years+1)
822     LCClower=PVCCL2(length,years+1)+1;
823 else
824     LCClower=PVCCL3(length,years+1)+1;
825 end;
826 LCCmedian=LCClower-(((LCClower-LCCupper)/10)*3);
827
828 %Calculate upper, median and lower values for Capacity KPI
829 Capacitylower=0;
830 Capacityupper=0;
831 if Capacity1(1,1)<Capacity2(1,1) & Capacity1(1,1)<Capacity3(1,1)
832     Capacityupper=Capacity1(1,1)-1;
833 elseif Capacity2(1,1)<Capacity3(1,1)
834     Capacityupper=Capacity2(1,1)-1;
835 else
836     Capacityupper=Capacity3(1,1)-1;
837 end;
838
839 if Capacity1(1,1)>Capacity2(1,1) & Capacity1(1,1)>Capacity3(1,1)
840     Capacitylower=Capacity1(1,1)+1;
841 elseif Capacity2(1,1)>Capacity3(1,1)
842     Capacitylower=Capacity2(1,1)+1;
843 else;
844     Capacitylower=Capacity3(1,1)+1;
845 end;
846 Capacitymedian=Capacitylower-(((Capacitylower-Capacityupper)/10)*3);
847
848 %Calculate median values for SIL KPI
849 SILmedian=SILlower-(((SILlower-SILupper)/10)*3);
850
851 %Calculate upper, median and lower values for Voltage Regulation KPI
852 VRupper=0;
853 VRlower=0.50;
854 VRmedian=VRlower-(((VRlower-VRupper)/10)*3);
855
856 h = waitbar(0,'Calculating ATIs...');
857
858 for i=1:length
859     waitbar(i/length
860         for j=1:years+1
861             LCCc1(i,j)=10-(PVCCL1(i,j)-LCCupper)/((LCClower-LCCupper)/10);
862             LCCc2(i,j)=10-(PVCCL2(i,j)-LCCupper)/((LCClower-LCCupper)/10);
863             LCCc3(i,j)=10-(PVCCL3(i,j)-LCCupper)/((LCClower-LCCupper)/10);
864
865             Capacity1(i,j)=10-(Capacity1(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
866             Capacity2(i,j)=10-(Capacity2(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
867             Capacity3(i,j)=10-(Capacity3(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
868
869             SILc1(i,j)=10-(SILc1(i,j)-SILupper)/((SILlower-SILupper)/10);
870             SILc2(i,j)=10-(SILc2(i,j)-SILupper)/((SILlower-SILupper)/10);
871             SILc3(i,j)=10-(SILc3(i,j)-SILupper)/((SILlower-SILupper)/10);
872
873             %Especially only for shunt compensation
874             if comp1==1
875                 VRc1(i,j)=10;
876             else
877                 VRc1(i,j)=10-((abs(Vreg1(i,j)/100))-VRupper)/((VRlower-VRupper)/10);
878             end;
879             if comp2==1
880                 VRc2(i,j)=10;
881             else
882                 VRc2(i,j)=10-((abs(Vreg2(i,j)/100))-VRupper)/((VRlower-VRupper)/10);
883             end;
884             if comp3==1
885                 VRc3(i,j)=10;
886             else
887                 VRc3(i,j)=10-((abs(Vreg3(i,j)/100))-VRupper)/((VRlower-VRupper)/10);
888             end;
889
890             ATI1(i,j)=LCCc1(i,j)*w1 + Capacity1(i,j)*w2 + SILc1(i,j)*w3 + VRc1(i,j)*w4;
891             ATI2(i,j)=LCCc2(i,j)*w1 + Capacity2(i,j)*w2 + SILc2(i,j)*w3 + VRc2(i,j)*w4;
892             ATI3(i,j)=LCCc3(i,j)*w1 + Capacity3(i,j)*w2 + SILc3(i,j)*w3 + VRc3(i,j)*w4;
893
894             %Determine minimum and maximum values for ATI contour plot
895             if (ati==1) & (ATI1(i,j)>contourmax1)
896                 contourmax1=ATI1(i,j);
897             end;
898             if (ati==1) & (ATI2(i,j)>contourmax2)
899                 contourmax2=ATI2(i,j);

```



```

900         end;
901         if (ati==1) & (AT13(i,j)>contourmax3)
902             contourmax3=AT13(i,j);
903         end;
904
905         if (ati==1) & (AT11(i,j)<contourmin1)
906             contourmin1=AT11(i,j);
907         end;
908         if (ati==1) & (AT12(i,j)<contourmin2)
909             contourmin2=AT12(i,j);
910         end;
911         if (ati==1) & (AT13(i,j)<contourmin3)
912             contourmin3=AT13(i,j);
913         end;
914
915     end;
916 end;
917
918 close(h);
919
920 %Determine if contour plot is valid for ATIs of conductor 1,2 and 3
921 if (ati==1) & (contour>=contourmin1) & (contour<=contourmax1)
922     plotc1cont=1;
923 end;
924 if (ati==1) & (contour>=contourmin2) & (contour<=contourmax2)
925     plotc2cont=1;
926 end;
927 if (ati==1) & (contour>=contourmin3) & (contour<=contourmax3)
928     plotc3cont=1;
929 end;
930
931 Data(:,1) = Vreg1;
932 Data(:,2) = Vreg2;
933 Data(:,3) = Vreg3;
934 Data(:,4) = Ploss1;
935 Data(:,5) = Ploss2;
936 Data(:,6) = Ploss3;
937 Data(:,7) = PVCCL1;
938 Data(:,8) = PVCCL2;
939 Data(:,9) = PVCCL3;
940 Data(:,10) = AT11;
941 Data(:,11) = AT12;
942 Data(:,12) = AT13;
943
944 XData123(:,1) = XData1;
945 XData123(:,2) = XData2;
946 XData123(:,3) = XData3;
947
948 XD(:,1) = XD1;
949 XD(:,2) = XD2;
950 XD(:,3) = XD3;
951
952 LR(:,1) = LR1;
953 LR(:,2) = LR2;
954 LR(:,3) = LR3;

```

## B.4.2 LFDC.m

```

1 function lfdc(action)
2 %Main Loadflow for dc bipole line GUI
3
4 switch(action)
5
6 case 'VoltageRegulation'
7     set(gcbo,'Value',1);
8     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','%');
9     set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
10    set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
11    set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATlrb'),'Value',0);
12    set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
13    set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');
14
15 case 'Loss'
16     set(gcbo,'Value',1);
17     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','kW');
18     set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
19     set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
20     set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATlrb'),'Value',0);
21     set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
22     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','10');
23
24 case 'CCL'
25     set(gcbo,'Value',1);
26     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','R');
27     set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
28     set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
29     set(findobj(findobj('Tag','lfdcGUI'),'Tag','ATlrb'),'Value',0);
30     set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
31     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','100000');
32
33 case 'ATT'
34     set(gcbo,'Value',1);
35     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcontunit'),'String','');
36     set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value',0);
37     set(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value',0);
38     set(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value',0);
39     set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
40     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');
41
42 case 'Posslider'
43     length=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','lengthe'),'String'));
44     UserData=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData');
45     set(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Max',length);
46     set(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Sliderstep',[1/length 10/length]);
47     set(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Min',1);
48     position=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Value'));
49     if position>length
50         set(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Value',1);
51         set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String','1');
52         UserData(27)=1;
53     else
54         set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String',num2str(position));
55         UserData(27)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','possider'),'Value'));
56     end;
57     set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData',UserData);
58     Change2dPlotsDC([1 0 0])
59
60 case 'Yearslider'
61     years=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearse'),'String'));
62     UserData=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData');
63     set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Max',years);
64     set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Sliderstep',[1/years 10/years]);
65     set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Min',0);
66     year=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Value'));
67     if year>years
68         set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Value',0);
69         set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String','0');
70         UserData(28)=0;
71     else
72         set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String',num2str(year));
73         UserData(28)=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearsider'),'Value'));
74     end;
75     set(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'UserData',UserData);
76     Change2dPlotsDC([0 1 0])
77
78 case 'Contourslider'
79     vr=get(findobj(findobj('Tag','lfdcGUI'),'Tag','vrrb'),'Value');
80     loss=get(findobj(findobj('Tag','lfdcGUI'),'Tag','lossrb'),'Value');
81     ccl=get(findobj(findobj('Tag','lfdcGUI'),'Tag','cclrb'),'Value');
82     ati=get(findobj(findobj('Tag','lfdcGUI'),'Tag','ATlrb'),'Value');
83     if vr==1
84         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',100);
85         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
86         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
87         valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
88         if valuecontslider>100
89             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
90         end;
91     set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
92     elseif loss==1
93         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
94         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
95         set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
96         valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));

```



```

97         if valueconslider==0
98             set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value',10);
99         else
100             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcnt'),'String',num2str(valueconslider));
101         end;
102     elseif ccl==1
103         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Max',10e3);
104         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Sliderstep',[0.001 0.01]);
105         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Min',0);
106         valueconslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value'));
107         if valueconslider==0
108             set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value',10);
109         else;
110             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcnt'),'String',num2str(valueconslider*10000));
111         end;
112     elseif ati==1
113         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Max',10);
114         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Sliderstep',[0.11111111 0.5]);
115         set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Min',1);
116         valueconslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value'));
117         if valueconslider==0
118             set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value',1);
119         elseif valueconslider>10
120             set(findobj(findobj('Tag','lfdcGUI'),'Tag','conslider'),'Value',1);
121         else
122             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcnt'),'String',num2str(valueconslider));
123         end;
124     end;
125
126 case 'r1'
127     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
128 case 'r2'
129     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
130 case 'r3'
131     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
132 case 'sysV'
133     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
134 case 'Vs'
135     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
136 case 'length'
137     oldvalue=get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max');
138     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
139     newvalue=str2num(get(gcbo,'String'));
140     if newvalue<=1
141         set(gcbo,'String',oldvalue);
142     end;
143 case 'LG'
144     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
145 case 'IL'
146     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
147 case 'years'
148     oldvalue=get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max');
149     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
150     newvalue=str2num(get(gcbo,'String'));
151     if newvalue<=0
152         set(gcbo,'String',oldvalue);
153     elseif newvalue>25
154         set(gcbo,'String',oldvalue);
155     end;
156 case 'stationcost'
157     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
158 case 'stationloss'
159     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
160 case 'interest'
161     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
162 case 'ALF'
163     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
164 case 'ampc1'
165     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
166 case 'ampc2'
167     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
168 case 'ampc3'
169     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
170 case 'w1'
171     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
172     value=str2num(get(gcbo,'String'));
173     if value>1
174         set(gcbo,'String','');
175     end;
176 case 'w2'
177     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
178     value=str2num(get(gcbo,'String'));
179     if value>1
180         set(gcbo,'String','');
181     end;
182 case 'w4'
183     set(gcbo,'String',num2str(str2num(get(gcbo,'String')))) %erase input on error
184     value=str2num(get(gcbo,'String'));
185     if value>1
186         set(gcbo,'String','');
187     end;
188
189 case 'Run'
190     %Initialize variables
191     clear all;
192     plotc1cont=[];
193     plotc2cont=[];
194     plotc3cont=[];
195
196     GC=str2num(get(findobj(findobj('Tag','lfdcGUI'),'Tag','gclb'),'String'));
197     namec1=get(findobj(findobj('Tag','lfdcGUI'),'Tag','cle'),'String'); %Get name of conductor 1

```

```

198 namec2=get(findobj(findobj('Tag','lfdcGUI','Tag','c2e'),'String')); %Get name of conductor 2
199 namec3=get(findobj(findobj('Tag','lfdcGUI','Tag','c3e'),'String')); %Get name of conductor 3
200
201 UserData(1)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','r1e'),'String'))); %resistance/km of conductor 1
202 UserData(2)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','r2e'),'String'))); %resistance/km of conductor 2
203 UserData(3)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','r3e'),'String'))); %resistance/km of conductor 3
204 UserData(4)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','costc1e'),'String')));
205 UserData(5)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','costc2e'),'String')));
206 UserData(6)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','costc3e'),'String')));
207 UserData(7)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','ampc1e'),'String')));
208 UserData(8)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','ampc2e'),'String')));
209 UserData(9)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','ampc3e'),'String')));
210 UserData(10)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','sysve'),'String'))*1e3); %Get system voltage [kV]
211 UserData(11)=(str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','sendve'),'String'))/100)*UserData(10));
212 UserData(12)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','lengthe'),'String'))); %Get line length [km]
213 UserData(13)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','lge'),'String'))/100); %Get load growth [%]
214 UserData(14)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','ile'),'String'))*1e6); %Get initial load [MW]
215 UserData(15)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','yarse'),'String'))); %Get years - normally 25
216 UserData(16)=(str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','inte'),'String'))/100);
217 UserData(17)=(str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','alfe'),'String'))^2);
218 UserData(18)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','ccse'),'String')));
219 UserData(19)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','cle'),'String'))/100);
220 UserData(20)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','w1e'),'String')));
221 UserData(21)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','w2e'),'String')));
222 UserData(22)=str2num(get(findobj(findobj('Tag','lfdcGUI','Tag','w4e'),'String')));
223 UserData(23)=get(findobj(findobj('Tag','lfdcGUI','Tag','vrrb'),'Value'));
224 UserData(24)=get(findobj(findobj('Tag','lfdcGUI','Tag','lossrb'),'Value'));
225 UserData(25)=get(findobj(findobj('Tag','lfdcGUI','Tag','cclrb'),'Value'));
226 UserData(26)=get(findobj(findobj('Tag','lfdcGUI','Tag','ATlrb'),'Value'));
227
228 SetSliders(UserData);
229
230 UserData(27)=round(get(findobj(findobj('Tag','lfdcGUI','Tag','posslder'),'Value')));
231 UserData(28)=round(get(findobj(findobj('Tag','lfdcGUI','Tag','yearslder'),'Value')));
232 UserData(29)=round(get(findobj(findobj('Tag','lfdcGUI','Tag','contsllder'),'Value')));
233
234 %Calculate voltage regulation,loss, ccl and ati matrices
235 [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
236 =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont);
237
238 %3d plots of voltage regulation, loss and ati plots
239 plot3d(XData123,YData,namec1,namec2,namec3,UserData,Data)
240
241 %2d plots of cross-section through 3d voltage regulation, loss and ati plots
242 plot2d(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)
243
244 set(findobj(findobj('Tag','lfdcGUI','Tag','vrrb'),'UserData',UserData),
245 set(findobj(findobj('Tag','lfdcGUI','Tag','lossrb'),'UserData',Data),
246 set(findobj(findobj('Tag','lfdcGUI','Tag','cclrb'),'UserData',XData123),
247 set(findobj(findobj('Tag','lfdcGUI','Tag','ATlrb'),'UserData',YData),
248
249 set(findobj(findobj('Tag','lfdcGUI','Tag','rpb'),'UserData',XD),
250 set(findobj(findobj('Tag','lfdcGUI','Tag','resetpb'),'UserData',XData),
251 set(findobj(findobj('Tag','lfdcGUI','Tag','closepb'),'UserData',LR),
252
253 set(findobj(findobj('Tag','lfdcGUI','Tag','c1e'),'UserData',plotc1cont);
254 set(findobj(findobj('Tag','lfdcGUI','Tag','c2e'),'UserData',plotc2cont);
255 set(findobj(findobj('Tag','lfdcGUI','Tag','c3e'),'UserData',plotc3cont);
256
257 case 'reset'
258 set(findobj(findobj('Tag','lfdcGUI','Tag','vrrb'),'Value',1);
259 set(findobj(findobj('Tag','lfdcGUI','Tag','lossrb'),'Value',0);
260 set(findobj(findobj('Tag','lfdcGUI','Tag','cclrb'),'Value',0);
261 set(findobj(findobj('Tag','lfdcGUI','Tag','ATlrb'),'Value',0);
262
263 set(findobj(findobj('Tag','lfdcGUI','Tag','posslder'),'Value',1);
264 set(findobj(findobj('Tag','lfdcGUI','Tag','posslder'),'Max',10);
265 set(findobj(findobj('Tag','lfdcGUI','Tag','Dispos'),'String','1');
266 set(findobj(findobj('Tag','lfdcGUI','Tag','yearslder'),'Value',0);
267 set(findobj(findobj('Tag','lfdcGUI','Tag','yearslder'),'Max',25);
268 set(findobj(findobj('Tag','lfdcGUI','Tag','Dispyear'),'String','0');
269 set(findobj(findobj('Tag','lfdcGUI','Tag','contsllder'),'Value',1);
270 set(findobj(findobj('Tag','lfdcGUI','Tag','contsllder'),'Max',100);
271 set(findobj(findobj('Tag','lfdcGUI','Tag','Dispcnt'),'String','1');
272
273 set(findobj(findobj('Tag','lfdcGUI','Tag','c1e'),'String','');
274 set(findobj(findobj('Tag','lfdcGUI','Tag','r1e'),'String','');
275 set(findobj(findobj('Tag','lfdcGUI','Tag','costc1e'),'String','');
276 set(findobj(findobj('Tag','lfdcGUI','Tag','ampc1e'),'String','');
277 set(findobj(findobj('Tag','lfdcGUI','Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]));
278 set(findobj(findobj('Tag','lfdcGUI','Tag','c2e'),'String','');
279 set(findobj(findobj('Tag','lfdcGUI','Tag','r2e'),'String','');
280 set(findobj(findobj('Tag','lfdcGUI','Tag','costc2e'),'String','');
281 set(findobj(findobj('Tag','lfdcGUI','Tag','ampc2e'),'String','');
282 set(findobj(findobj('Tag','lfdcGUI','Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]));
283 set(findobj(findobj('Tag','lfdcGUI','Tag','c3e'),'String','');
284 set(findobj(findobj('Tag','lfdcGUI','Tag','r3e'),'String','');
285 set(findobj(findobj('Tag','lfdcGUI','Tag','costc3e'),'String','');
286 set(findobj(findobj('Tag','lfdcGUI','Tag','ampc3e'),'String','');
287 set(findobj(findobj('Tag','lfdcGUI','Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]));
288 set(findobj(findobj('Tag','lfdcGUI','Tag','sysve'),'String','');
289 set(findobj(findobj('Tag','lfdcGUI','Tag','sendve'),'String','');
290 set(findobj(findobj('Tag','lfdcGUI','Tag','lengthe'),'String','');
291 set(findobj(findobj('Tag','lfdcGUI','Tag','lge'),'String','');
292 set(findobj(findobj('Tag','lfdcGUI','Tag','ile'),'String','');
293 set(findobj(findobj('Tag','lfdcGUI','Tag','yarse'),'String','');
294 set(findobj(findobj('Tag','lfdcGUI','Tag','inte'),'String','');
295 set(findobj(findobj('Tag','lfdcGUI','Tag','gclb'),'Value',1);
296 set(findobj(findobj('Tag','lfdcGUI','Tag','alfe'),'String','');
297 set(findobj(findobj('Tag','lfdcGUI','Tag','ccse'),'String','');
298 set(findobj(findobj('Tag','lfdcGUI','Tag','w1e'),'String','');

```



```

299     set(findobj(findobj('Tag','lfdcGUI'),'Tag','w2e'),'String','');
300     set(findobj(findobj('Tag','lfdcGUI'),'Tag','w4e'),'String','');
301     set(findobj(findobj('Tag','lfdcGUI'),'Tag','cle'),'String','');
302
303     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl1axes'))
304     cla reset
305     set(gca,'Tag','vrl1axes');
306     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl2axes'))
307     cla reset
308     set(gca,'Tag','vrl2axes');
309     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl3axes'))
310     cla reset
311     set(gca,'Tag','vrl3axes');
312     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlposaxes'))
313     cla reset
314     set(gca,'Tag','vrlposaxes');
315     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlyearaxes'))
316     cla reset
317     set(gca,'Tag','vrlyearaxes');
318     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
319     cla reset
320     set(gca,'Tag','vrlcontaxes');
321
322     if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
323         set(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value',0)
324         fig = findobj('Tag','3dvrlc1fig');
325         if ~isempty(fig)
326             close.figure(fig);
327         end;
328         fig = findobj('Tag','3dvrlc2fig');
329         if ~isempty(fig)
330             close.figure(fig);
331         end;
332         fig = findobj('Tag','3dvrlc3fig');
333         if ~isempty(fig)
334             close.figure(fig);
335         end;
336         fig = findobj('Tag','2dvrlpointloadfig');
337         if ~isempty(fig)
338             close.figure(fig);
339         end;
340         fig = findobj('Tag','2dvrlposfig');
341         if ~isempty(fig)
342             close.figure(fig);
343         end;
344         fig = findobj('Tag','2dvrlcontfig');
345         if ~isempty(fig)
346             close.figure(fig);
347         end;
348     end;
349
350     case 'CloseHighResolutionFigures'
351         fig = findobj('Tag','3dvrlc1fig');
352         if ~isempty(fig)
353             close.figure(fig);
354         end;
355         fig = findobj('Tag','3dvrlc2fig');
356         if ~isempty(fig)
357             close.figure(fig);
358         end;
359         fig = findobj('Tag','3dvrlc3fig');
360         if ~isempty(fig)
361             close.figure(fig);
362         end;
363         fig = findobj('Tag','2dvrlpointloadfig');
364         if ~isempty(fig)
365             close.figure(fig);
366         end;
367         fig = findobj('Tag','2dvrlposfig');
368         if ~isempty(fig)
369             close.figure(fig);
370         end;
371         fig = findobj('Tag','2dvrlcontfig');
372         if ~isempty(fig)
373             close.figure(fig);
374         end;
375
376     case 'close'
377         if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
378             set(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value',0)
379             fig = findobj('Tag','3dvrlc1fig');
380             if ~isempty(fig)
381                 close.figure(fig);
382             end;
383             fig = findobj('Tag','3dvrlc2fig');
384             if ~isempty(fig)
385                 close.figure(fig);
386             end;
387             fig = findobj('Tag','3dvrlc3fig');
388             if ~isempty(fig)
389                 close.figure(fig);
390             end;
391             fig = findobj('Tag','2dvrlpointloadfig');
392             if ~isempty(fig)
393                 close.figure(fig);
394             end;
395             fig = findobj('Tag','2dvrlposfig');
396             if ~isempty(fig)
397                 close.figure(fig);
398             end;
399             fig = findobj('Tag','2dvrlcontfig');

```

```

400         if ~isempty(fig)
401             close(fgure(fig));
402         end;
403     end;
404     fig = findobj('Tag','lfdcGUI');
405     saveas(fgure(fig),get(fig,'FileName'),'fig')
406     close(fgure(fig));
407
408     function SetSliders(UserData)
409         length=UserData(12);
410         years=UserData(15);
411         vr=UserData(23);
412         loss=UserData(24);
413         ccl=UserData(25);
414         ati=UserData(26);
415
416         set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Max',length);
417         set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Sliderstep',[1/length 10/length]);
418         set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Min',1);
419         position=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value'));
420         if position>length
421             set(findobj(findobj('Tag','lfdcGUI'),'Tag','posslider'),'Value',1);
422             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String','1');
423         else
424             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Disppos'),'String',num2str(position));
425         end;
426
427         set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Max',years);
428         set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Sliderstep',[1/years 10/years]);
429         set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Min',0);
430         valueyearslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value'));
431         if valueyearslider>years
432             set(findobj(findobj('Tag','lfdcGUI'),'Tag','yearslider'),'Value',0);
433             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String','0');
434         else
435             set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispyear'),'String',num2str(valueyearslider));
436         end;
437
438         if vr==1
439             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',100);
440             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.01 0.1]);
441             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
442             valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
443             if valuecontslider>100
444                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
445                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','1');
446             else
447                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
448             end;
449         elseif loss==1
450             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
451             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
452             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
453             valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
454             if valuecontslider==0
455                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
456                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','10');
457             else
458                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
459             end;
460         elseif ccl==1
461             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10e3);
462             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.001 0.01]);
463             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',0);
464             valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
465             if valuecontslider==0
466                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',10);
467                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String','100000');
468             else
469                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider*100000));
470             end;
471         elseif ati==1
472             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Max',10);
473             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Sliderstep',[0.11111111 0.5]);
474             set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Min',1);
475             valuecontslider=round(get(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value'));
476             if valuecontslider==0
477                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
478             elseif valuecontslider>10
479                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','contslider'),'Value',1);
480             else
481                 set(findobj(findobj('Tag','lfdcGUI'),'Tag','Dispcont'),'String',num2str(valuecontslider));
482             end;
483         end;
484
485     function [Data,XData,XData123,LR,XD,YData,plotc1cont,plotc2cont,plotc3cont]
486         =VRandLoss(UserData,GC,plotc1cont,plotc2cont,plotc3cont)
487
488         rc1=UserData(1);
489         rc2=UserData(2);
490         rc3=UserData(3);
491         costc1=UserData(4);
492         costc2=UserData(5);
493         costc3=UserData(6);
494         ampc1=UserData(7);
495         ampc2=UserData(8);
496         ampc3=UserData(9);
497         sysV=UserData(10);
498         Vs=UserData(11);
499         length=UserData(12);
500         LG=UserData(13);
501         IL=UserData(14);

```



```

501 years=UserData(15);
502 Interest=UserData(16);
503 ELF=UserData(17);
504 stationcost=UserData(18);
505 stationloss=UserData(19);
506 w1=UserData(20);
507 w2=UserData(21);
508 w4=UserData(22);
509 vr=UserData(23);
510 loss=UserData(24);
511 ccl=UserData(25);
512 ati=UserData(26);
513 position=UserData(27);
514 year=UserData(28);
515 contour=UserData(29);
516
517 Pbase=100e6; %Power base - usually chosen as 100 MW
518 Zbase=(sysV^2)/Pbase; %Calculate Zbase from Pbase and SysV
519
520 Vspu=Vs/sysV; %Calculate the sending voltage in p.u.
521
522 XD1=zeros([length,1]);
523 XD2=zeros([length,1]);
524 XD3=zeros([length,1]);
525
526 testLR1=zeros([(years+1),1]);
527 testLR2=zeros([(years+1),1]);
528 testLR3=zeros([(years+1),1]);
529
530 for c=1:(years+1)
531     XData(c)=(IL/1e6)*(1+LG)^(c-1); %Set up XData vector for 3d plot
532     DF(c)=(1+Interest)^(c-1); %Discount Factor
533     CLA(c)=8760*(GC(c)/100)*ELF; %Cost of losses/kW/Annum [R]
534     CL(c)=(IL/1000)*(2*stationloss)*(1+LG)^(c-1); %Converter losses [kW]
535 end;
536
537 CCS=2*stationcost*((IL*(1+LG)^(years))/1e6); %Cost of two converter stations [R]
538 contourmax1=0;
539 contourmax2=0;
540 contourmax3=0;
541 contourmin1=10;
542 contourmin2=10;
543 contourmin3=10;
544
545 h = waitbar(0,'Calculating matrices...');
546
547 YData=1:length;
548 for j=1:(years+1)
549     LR1(j)=length;
550     LR2(j)=length;
551     LR3(j)=length;
552 end;
553
554 for i=1:length
555     waitbar(i/length)
556
557     R1pu=(rc1*i)/Zbase; %Total p.u. resistance for conductor 1
558     R2pu=(rc2*i)/Zbase; %Total p.u. resistance for conductor 2
559     R3pu=(rc3*i)/Zbase; %Total p.u. resistance for conductor 3
560
561     XData1(i,:)=XData;
562     XData2(i,:)=XData;
563     XData3(i,:)=XData;
564
565     for j=1:(years+1)
566
567         Prpu=(IL/Pbase)*(1+LG)^(j-1); %Receiving end power in p.u. for year(j-1)
568         Vr1pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R1pu*Prpu))/2; %Percentage Voltage Regulation for conductor 1
569         Vr2pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R2pu*Prpu))/2; %Percentage Voltage Regulation for conductor 2
570         Vr3pu(i,j)=(Vspu+sqrt((-Vspu)^2-4*0.5*R3pu*Prpu))/2; %Percentage Voltage Regulation for conductor 3
571
572         if isreal(Vr1pu(i,j))
573             Vreg1(i,j)=(Vspu-abs(Vr1pu(i,j)))*100; %voltage regulation for conductor 1
574             Ps1(i,j)=Vspu/(abs(Vr1pu(i,j)))*Prpu*Pbase; %Sending end power of conductor 1
575             Ploss1(i,j)=(Ps1(i,j)-Prpu*Pbase)/1000+CL(j); %Losses in kW of conductor 1
576             ECL1(i,j)=Ploss1(i,j)*CLA(j); %Escalated cost of losses [R] of conductor 1
577             PVCL1(i,j)=ECL1(i,j)/DF(j); %Present Value cost of losses of conductor 1
578             PVCL1(:,1)=0; %Present value of year 0 = 0
579             jrepeat1=j;
580         else
581             if testLR1(j)==0
582                 LR1(j)=i-1;
583                 testLR1(j)=1;
584             end;
585             Vreg1(i,j)=Vreg1(i,jrepeat1);
586             Ps1(i,j)=Ps1(i,jrepeat1);
587             Ploss1(i,j)=Ploss1(i,jrepeat1);
588             PVCL1(i,j)=0;
589             XData1(i,j)=XData(jrepeat1);
590             XD1(i)=1;
591         end;
592         if isreal(Vr2pu(i,j))
593             Vreg2(i,j)=(Vspu-abs(Vr2pu(i,j)))*100; %voltage regulation for conductor 2
594             Ps2(i,j)=Vspu/(abs(Vr2pu(i,j)))*Prpu*Pbase; %Sending end power of conductor 2
595             Ploss2(i,j)=(Ps2(i,j)-Prpu*Pbase)/1000+CL(j); %Losses in kW of conductor 2
596             ECL2(i,j)=Ploss2(i,j)*CLA(j); %Escalated cost of losses [R] of conductor 2
597             PVCL2(i,j)=ECL2(i,j)/DF(j); %Present Value cost of losses of conductor 2
598             PVCL2(:,1)=0;
599             jrepeat2=j;
600         else
601             if testLR2(j)==0

```

```

602         LR2(j)=i-1;
603         testLR2(j)=1;
604     end;
605     Vreg2(i,j)=Vreg2(i,jrepeat2);
606     XData2(i,j)=XData(jrepeat2);
607     Ps2(i,j)=Ps2(i,jrepeat2);
608     Ploss2(i,j)=Ploss2(i,jrepeat2);
609     PVCL2(i,j)=0;
610     XData2(i,j)=XData(jrepeat2);
611     XD2(i)=2;
612 end;
613 if isreal(Vr3pu(i,j))
614     Vreg3(i,j)=(Vspu-abs(Vr3pu(i,j)))*100; %voltage regulation for conductor 2
615     jrepeat3=j;
616     Ps3(i,j)=Vspu/(abs(Vr3pu(i,j)))*Prpu*Pbase; %Sending end power of conductor 3
617     Ploss3(i,j)=(Ps3(i,j)-Prpu*Pbase)/1000+CL(j); %Losses in kW of conductor 3
618     ECL3(i,j)=Ploss3(i,j)*CLA(j); %Escalated cost of losses [R] of conductor 3
619     PVCL3(i,j)=ECL3(i,j)/DF(j); %Present Value cost of losses of conductor 3
620     PVCL3(:,1)=0;
621 else
622     if testLR3(j)==0
623         LR3(j)=i-1;
624         testLR3(j)=1;
625     end;
626     Vreg3(i,j)=Vreg3(i,jrepeat3);
627     XData3(i,j)=XData(jrepeat3);
628     Ps3(i,j)=Ps3(i,jrepeat3);
629     Ploss3(i,j)=Ploss3(i,jrepeat3);
630     PVCL3(i,j)=0;
631     XData3(i,j)=XData(jrepeat3);
632     XD3(i)=1;
633 end;
634
635 if (vr==1) & (Vreg1(i,j)>=contour) %Determine if conductor 1 contour plot is valid
636     plotc1cont=1;
637 end;
638
639 if (vr==1) & (Vreg2(i,j)>=contour) %Determine if conductor 2 contour plot is valid
640     plotc2cont=1;
641 end;
642
643 if (vr==1) & (Vreg3(i,j)>=contour) %Determine if conductor 3 contour plot is valid
644     plotc3cont=1;
645 end;
646
647 Capacityc1(i,j)=(CCS+costc1*i)/(2*sysV*1e-3*ampc1*1*i); %Capacity Matrix of conductor 1
648 Capacityc2(i,j)=(CCS+costc2*i)/(2*sysV*1e-3*ampc2*1*i); %Capacity Matrix of conductor 2
649 Capacityc3(i,j)=(CCS+costc3*i)/(2*sysV*1e-3*ampc3*1*i); %Capacity Matrix of conductor 3
650
651 end;
652
653 CPVCL1(i,:)=cumsum(PVCL1(i,:)); %Cumulative Present Value Cost of Losses for Conductor 1
654 CPVCL2(i,:)=cumsum(PVCL2(i,:)); %Cumulative Present Value Cost of Losses for Conductor 2
655 CPVCL3(i,:)=cumsum(PVCL3(i,:)); %Cumulative Present Value Cost of Losses for Conductor 3
656
657 PVCCL1(i,:)=CPVCL1(i,:)+costc1*i; %Cumulative Present Value of Capital and Cost of Losses
658 PVCCL2(i,:)=CPVCL2(i,:)+costc2*i; %Cumulative Present Value of Capital and Cost of Losses
659 PVCCL3(i,:)=CPVCL3(i,:)+costc3*i; %Cumulative Present Value of Capital and Cost of Losses
660
661 PVCCL1(i,:)=(PVCCL1(i,:)+CCS)/1e6; %Add cost of two converter stations
662 PVCCL2(i,:)=(PVCCL2(i,:)+CCS)/1e6;
663 PVCCL3(i,:)=(PVCCL3(i,:)+CCS)/1e6;
664
665 for n=1:(years+1)
666     if (ccl==1) & (PVCCL1(i,n)>=contour*10000) & (contour*10000>PVCCL1(1,1))
667         plotc1cont=1;
668     elseif (loss==1) & (Ploss1(i,n)>=contour) & (contour>Ploss1(1,1))
669         plotc1cont=1;
670     end;
671     if (ccl==1) & (PVCCL2(i,n)>=contour*10000) & (contour*10000>PVCCL2(1,1))
672         plotc2cont=1;
673     elseif (loss==1) & (Ploss2(i,n)>=contour) & (contour>Ploss2(1,1))
674         plotc2cont=1;
675     end;
676     if (ccl==1) & (PVCCL3(i,n)>=contour*10000) & (contour*10000>PVCCL3(1,1))
677         plotc3cont=1;
678     elseif (loss==1) & (Ploss3(i,n)>=contour) & (contour>Ploss3(1,1))
679         plotc3cont=1;
680     end;
681 end;
682 end;
683
684 close(h);
685
686 if isreal(Vr1pu)
687     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1t'),'BackgroundColor',[.8314 .8157 .7843]);
688 else
689     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c1t'),'BackgroundColor','r');
690 end;
691 if isreal(Vr2pu)
692     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2t'),'BackgroundColor',[.8314 .8157 .7843]);
693 else
694     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c2t'),'BackgroundColor','r');
695 end;
696 if isreal(Vr3pu)
697     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3t'),'BackgroundColor',[.8314 .8157 .7843]);
698 else
699     set(findobj(findobj('Tag','lfdcGUI'),'Tag','c3t'),'BackgroundColor','r');
700 end;
701
702 %Calculate upper, median and lower values for LCC KPI

```



```

703 LCClower=0;
704 LCCupper=CCS;
705 if PVCCL1(length,years+1)>PVCCL2(length,years+1) & PVCCL1(length,years+1)>PVCCL3(length,years+1)
706     LCClower=PVCCL1(length,years+1)*1e6+1e6;
707 elseif PVCCL2(length,years+1)>PVCCL3(length,years+1)
708     LCClower=PVCCL2(length,years+1)*1e6+1e6;
709 else
710     LCClower=PVCCL3(length,years+1)*1e6+1e6;
711 end;
712 LCCmedian=LCClower-(((LCClower-LCCupper)/10)*3);
713
714 %Calculate upper, median and lower values for Capacity KPI
715 Capacitylower=0;
716 Capacityupper=0;
717 if Capacityc1(length,1)<Capacityc2(length,1) & Capacityc1(length,1)<Capacityc3(length,1)
718     Capacityupper=Capacityc1(length,1)-1;
719 elseif Capacityc2(length,1)<Capacityc3(length,1)
720     Capacityupper=Capacityc2(length,1)-1;
721 else
722     Capacityupper=Capacityc3(length,1)-1;
723 end;
724 if Capacityc1(1,1)>Capacityc2(1,1) & Capacityc1(1,1)>Capacityc3(1,1)
725     Capacitylower=Capacityc1(1,1)+1;
726 elseif Capacityc2(1,1)>Capacityc3(1,1)
727     Capacitylower=Capacityc2(1,1)+1;
728 else;
729     Capacitylower=Capacityc3(1,1)+1;
730 end;
731 Capacitymedian=Capacitylower-(((Capacitylower-Capacityupper)/10)*3);
732
733 %Calculate upper, median and lower values for Voltage Regulation KPI
734 VRupper=0.000001;
735 VRlower=0.5;
736 VRmedian=VRlower-(((VRlower-VRupper)/10)*3);
737
738 h = waitbar(0,'Calculating ATIs...');
739
740 for i=1:length
741     waitbar(i/length)
742     for j=1:years+1
743         LCCc1(i,j)=10-(PVCCL1(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);
744         LCCc2(i,j)=10-(PVCCL2(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);
745         LCCc3(i,j)=10-(PVCCL3(i,j)*1e6-LCCupper)/((LCClower-LCCupper)/10);
746
747         Capacityc1(i,j)=10-(Capacityc1(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
748         Capacityc2(i,j)=10-(Capacityc2(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
749         Capacityc3(i,j)=10-(Capacityc3(i,j)-Capacityupper)/((Capacitylower-Capacityupper)/10);
750
751         VRc1(i,j)=10-((Vreg1(i,j)/100)-VRupper)/((VRlower-VRupper)/10);
752         VRc2(i,j)=10-((Vreg2(i,j)/100)-VRupper)/((VRlower-VRupper)/10);
753         VRc3(i,j)=10-((Vreg3(i,j)/100)-VRupper)/((VRlower-VRupper)/10);
754
755         ATI1(i,j)=LCCc1(i,j)*w1+Capacityc1(i,j)*w2+VRc1(i,j)*w4;
756         ATI2(i,j)=LCCc2(i,j)*w1+Capacityc2(i,j)*w2+VRc2(i,j)*w4;
757         ATI3(i,j)=LCCc3(i,j)*w1+Capacityc3(i,j)*w2+VRc3(i,j)*w4;
758
759         %Determine minimum and maximum values for ATI contour plot
760         if (ati==1) & (ATI1(i,j)>contourmax1)
761             contourmax1=ATI1(i,j);
762         end;
763         if (ati==1) & (ATI2(i,j)>contourmax2)
764             contourmax2=ATI2(i,j);
765         end;
766         if (ati==1) & (ATI3(i,j)>contourmax3)
767             contourmax3=ATI3(i,j);
768         end;
769
770         if (ati==1) & (ATI1(i,j)<contourmin1)
771             contourmin1=ATI1(i,j);
772         end;
773         if (ati==1) & (ATI2(i,j)<contourmin2)
774             contourmin2=ATI2(i,j);
775         end;
776         if (ati==1) & (ATI3(i,j)<contourmin3)
777             contourmin3=ATI3(i,j);
778         end;
779     end;
780 end;
781
782 close(h);
783 %Determine if contour plot is valid for ATIs of conductor 1,2 and 3
784 if (ati==1) & (contour>=contourmin1) & (contour<=contourmax1)
785     plotc1cont=1;
786 end;
787 if (ati==1) & (contour>=contourmin2) & (contour<=contourmax2)
788     plotc2cont=1;
789 end;
790 if (ati==1) & (contour>=contourmin3) & (contour<=contourmax3)
791     plotc3cont=1;
792 end;
793
794 Data(:,1) = Vreg1;
795 Data(:,2) = Vreg2;
796 Data(:,3) = Vreg3;
797 Data(:,4) = Ploss1;
798 Data(:,5) = Ploss2;
799 Data(:,6) = Ploss3;
800 Data(:,7) = PVCCL1;
801 Data(:,8) = PVCCL2;
802 Data(:,9) = PVCCL3;

```

```
804      Data(:,10) = ATI1;
805      Data(:,11) = ATI2;
806      Data(:,12) = ATI3;
807
808      XData123(:,1) = XData1;
809      XData123(:,2) = XData2;
810      XData123(:,3) = XData3;
811
812      XD(:,1) = XD1;
813      XD(:,2) = XD2;
814      XD(:,3) = XD3;
815
816      LR(:,1) = LR1;
817      LR(:,2) = LR2;
818      LR(:,3) = LR3;
```



## B.4.3 Plot2d.m

```

1 function plot2d(XData,XData123,LR,XD,YData,namec1,namec2,namec3,plotc1cont,plotc2cont,plotc3cont,UserData,Data)
2
3     vr      =   UserData(23);
4     loss    =   UserData(24);
5     ccl     =   UserData(25);
6     ati     =   UserData(26);
7     position =   UserData(27);
8     year    =   UserData(28);
9     contour =   UserData(29);
10
11     Vreg1   =   Data(:,1);
12     Vreg2   =   Data(:,2);
13     Vreg3   =   Data(:,3);
14     Ploss1  =   Data(:,4);
15     Ploss2  =   Data(:,5);
16     Ploss3  =   Data(:,6);
17     PVCCCL1 =   Data(:,7);
18     PVCCCL2 =   Data(:,8);
19     PVCCCL3 =   Data(:,9);
20     ATI1    =   Data(:,10);
21     ATI2    =   Data(:,11);
22     ATI3    =   Data(:,12);
23
24     XData1  =   XData123(:,1);
25     XData2  =   XData123(:,2);
26     XData3  =   XData123(:,3);
27
28     XD1     =   XD(:,1);
29     XD2     =   XD(:,2);
30     XD3     =   XD(:,3);
31
32     LR1     =   LR(:,1);
33     LR2     =   LR(:,2);
34     LR3     =   LR(:,3);
35
36     Vregpos1 =   Vreg1(position,:);
37     Vregpos2 =   Vreg2(position,:);
38     Vregpos3 =   Vreg3(position,:);
39     losspos1 =   Ploss1(position,:);
40     losspos2 =   Ploss2(position,:);
41     losspos3 =   Ploss3(position,:);
42     PVCCCLpos1 =   PVCCCL1(position,:);
43     PVCCCLpos2 =   PVCCCL2(position,:);
44     PVCCCLpos3 =   PVCCCL3(position,:);
45     ATIp1    =   ATI1(position,:);
46     ATIp2    =   ATI2(position,:);
47     ATIp3    =   ATI3(position,:);
48
49     XDatapos1 =   XData1(position,:);
50     XDatapos2 =   XData2(position,:);
51     XDatapos3 =   XData3(position,:);
52
53     YData1   =   1:LR1(year+1);
54     YData2   =   1:LR2(year+1);
55     YData3   =   1:LR3(year+1);
56
57     for i1=1:LR1(year+1);
58         Vreg1LR(i1)=Vreg1(i1,year+1);
59         loss1LR(i1)=Ploss1(i1,year+1);
60         PVCCCL1LR(i1)=PVCCCL1(i1,year+1);
61         ATI1LR(i1)=ATI1(i1,year+1);
62     end;
63     for i2=1:LR2(year+1);
64         Vreg2LR(i2)=Vreg2(i2,year+1);
65         loss2LR(i2)=Ploss2(i2,year+1);
66         PVCCCL2LR(i2)=PVCCCL2(i2,year+1);
67         ATI2LR(i2)=ATI2(i2,year+1);
68     end;
69     for i3=1:LR3(year+1);
70         Vreg3LR(i3)=Vreg3(i3,year+1);
71         loss3LR(i3)=Ploss3(i3,year+1);
72         PVCCCL3LR(i3)=PVCCCL3(i3,year+1);
73         ATI3LR(i3)=ATI3(i3,year+1);
74     end;
75
76     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlposaxes'))
77     cla;
78     if vr==1
79         if XD1(position)==0;
80             H=plot(XData,Vregpos1,'r');
81             hold on;
82             set(gca,'Tag','vrlposaxes');
83             set(H,'LineWidth',1);
84         else
85             H=plot(XDatapos1,Vregpos1,'r');
86             hold on;
87             set(gca,'Tag','vrlposaxes');
88             set(H,'LineWidth',1);
89         end;
90         if XD2(position)==0;
91             H=plot(XData,Vregpos2,'b');
92             hold on;
93             set(gca,'Tag','vrlposaxes');
94             set(H,'LineWidth',1);
95         else
96             H=plot(XDatapos2,Vregpos2,'b');

```

```

97         hold on;
98         set(gca,'Tag','vrlposaxes');
99         set(H,'LineWidth',1);
100     end;
101     if XD3(position)==0;
102         H=plot(XData,Vregpos3,'g');
103         hold off;
104         set(gca,'Tag','vrlposaxes');
105         set(H,'LineWidth',1);
106     else
107         H=plot(XDatapos3,Vregpos3,'g');
108         hold off;
109         set(gca,'Tag','vrlposaxes');
110         set(H,'LineWidth',1);
111     end;
112     ylabel('Voltage Regulation [%]');
113     title('Voltage Regulation vs Point Load');
114 elseif loss==1
115     if XD1(position)==0;
116         H=plot(XData,losspos1,'r');
117         hold on;
118         set(gca,'Tag','vrlposaxes');
119         set(H,'LineWidth',1);
120     else
121         H=plot(XDatapos1,losspos1,'r');
122         hold on;
123         set(gca,'Tag','vrlposaxes');
124         set(H,'LineWidth',1);
125     end;
126     if XD2(position)==0;
127         H=plot(XData,losspos2,'b');
128         hold on;
129         set(gca,'Tag','vrlposaxes');
130         set(H,'LineWidth',1);
131     else
132         H=plot(XDatapos2,losspos2,'b');
133         hold on;
134         set(gca,'Tag','vrlposaxes');
135         set(H,'LineWidth',1);
136     end;
137     if XD3(position)==0;
138         H=plot(XData,losspos3,'g');
139         hold off;
140         set(gca,'Tag','vrlposaxes');
141         set(H,'LineWidth',1);
142     else
143         H=plot(XDatapos3,losspos3,'g');
144         hold off;
145         set(gca,'Tag','vrlposaxes');
146         set(H,'LineWidth',1);
147     end;
148     ylabel('Power Losses [kW]');
149     title('Power Losses vs Point Load');
150 elseif ccl==1
151     if XD1(position)==0;
152         H=plot(XData,PVCCLpos1,'r');
153         hold on;
154         set(gca,'Tag','vrlposaxes');
155         set(H,'LineWidth',1);
156     else
157         H=plot(XDatapos1,PVCCLpos1,'r');
158         hold on;
159         set(gca,'Tag','vrlposaxes');
160         set(H,'LineWidth',1);
161     end;
162     if XD2(position)==0;
163         H=plot(XData,PVCCLpos2,'b');
164         hold on;
165         set(gca,'Tag','vrlposaxes');
166         set(H,'LineWidth',1);
167     else
168         H=plot(XDatapos2,PVCCLpos2,'b');
169         hold on;
170         set(gca,'Tag','vrlposaxes');
171         set(H,'LineWidth',1);
172     end;
173     if XD3(position)==0;
174         H=plot(XData,PVCCLpos3,'g');
175         hold off;
176         set(gca,'Tag','vrlposaxes');
177         set(H,'LineWidth',1);
178     else
179         H=plot(XDatapos3,PVCCLpos3,'g');
180         hold off;
181         set(gca,'Tag','vrlposaxes');
182         set(H,'LineWidth',1);
183     end;
184     ylabel('Cumulative Cost [MR]');
185     title('LCC vs Point Load');
186 elseif ati==1
187     if XD1(position)==0;
188         H=plot(XData,ATipos1,'r');
189         hold on;
190         set(gca,'Tag','vrlposaxes');
191         set(H,'LineWidth',1);
192     else
193         H=plot(XDatapos1,ATipos1,'r');
194         hold on;
195         set(gca,'Tag','vrlposaxes');
196         set(H,'LineWidth',1);
197     end;

```



```

198     if XD2(position)==0;
199         H=plot(XData,ATIp2,'b');
200         hold on;
201         set(gca,'Tag','vrlposaxes');
202         set(H,'LineWidth',1);
203     else
204         H=plot(XDatapos2,ATIp2,'b');
205         hold on;
206         set(gca,'Tag','vrlposaxes');
207         set(H,'LineWidth',1);
208     end;
209     if XD3(position)==0;
210         H=plot(XData,ATIp3,'g');
211         hold off;
212         set(gca,'Tag','vrlposaxes');
213         set(H,'LineWidth',1);
214     else
215         H=plot(XDatapos3,ATIp3,'g');
216         hold off;
217         set(gca,'Tag','vrlposaxes');
218         set(H,'LineWidth',1);
219     end;
220     ylabel('ATT');
221     title('ATT vs Point Load');
222 end;
223 set(gca,'Tag','vrlposaxes');
224 axis tight
225 a = axis;
226 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
227 xlabel('Point Load [MW]');
228 legend(namec1,namec2,namec3,0);
229 grid on;
230
231 if get(findobj(findobj('Tag','lfdGUI'),'Tag','hrfcb'),'Value')
232     fsize = 36;
233     fig = findobj('Tag','2dvrlpointloadfig');
234     if isempty(fig)
235         figure
236         if vr==1
237             set(gcf,'IntegerHandle','off',...
238                 'Name','Voltage Regulation vs Point Load',...
239                 'NumberTitle','off',...
240                 'Tag','2dvrlpointloadfig');
241         elseif loss==1
242             set(gcf,'IntegerHandle','off',...
243                 'Name','Power Losses vs Point Load',...
244                 'NumberTitle','off',...
245                 'Tag','2dvrlpointloadfig');
246         elseif ccl==1
247             set(gcf,'IntegerHandle','off',...
248                 'Name','LCC vs Point Load',...
249                 'NumberTitle','off',...
250                 'Tag','2dvrlpointloadfig');
251         elseif ati==1
252             set(gcf,'IntegerHandle','off',...
253                 'Name','ATT vs Point Load',...
254                 'NumberTitle','off',...
255                 'Tag','2dvrlpointloadfig');
256         end;
257     else
258         figure(fig)
259         if vr==1
260             set(gcf,'IntegerHandle','off',...
261                 'Name','Voltage Regulation vs Point Load',...
262                 'NumberTitle','off',...
263                 'Tag','2dvrlpointloadfig');
264         elseif loss==1
265             set(gcf,'IntegerHandle','off',...
266                 'Name','Power Losses vs Point Load',...
267                 'NumberTitle','off',...
268                 'Tag','2dvrlpointloadfig');
269         elseif ccl==1
270             set(gcf,'IntegerHandle','off',...
271                 'Name','LCC vs Point Load',...
272                 'NumberTitle','off',...
273                 'Tag','2dvrlpointloadfig');
274         elseif ati==1
275             set(gcf,'IntegerHandle','off',...
276                 'Name','ATT vs Point Load',...
277                 'NumberTitle','off',...
278                 'Tag','2dvrlpointloadfig');
279         end;
280     end;
281     set(gcf,'Position',[1 29 1024 672]);
282     %set(gca,'Position',[0.15 0.15 0.7 0.7]);
283     set(gca,'Position',[0.15 0.15 0.82 0.82]);
284     if vr==1
285         if XD1(position)==0;
286             H=plot(XData,Vregpos1,'r');
287             hold on;
288             set(H,'LineWidth',1.5);
289         else
290             H=plot(XDatapos1,Vregpos1,'r');
291             hold on;
292             set(H,'LineWidth',1.5);
293         end;
294     if XD2(position)==0;
295         H=plot(XData,Vregpos2,'b');
296         hold on;
297         set(H,'LineWidth',1.5);
298     else

```

```

299         H=plot(XDatapos2,Vregpos2,'b');
300         hold on;
301         set(H,'LineWidth',1.5);
302     end;
303     if XD3(position)==0;
304         H=plot(XData,Vregpos3,'g');
305         hold off;
306         set(H,'LineWidth',1.5);
307     else
308         H=plot(XDatapos3,Vregpos3,'g');
309         hold off;
310         set(H,'LineWidth',1.5);
311     end;
312     hy=ylabel('Voltage Regulation [%]');
313     ht=title('Voltage Regulation vs Point Load');
314 elseif loss==1
315     if XD1(position)==0;
316         H=plot(XData,losspos1,'r');
317         hold on;
318         set(H,'LineWidth',1.5);
319     else
320         H=plot(XDatapos1,losspos1,'r');
321         hold on;
322         set(H,'LineWidth',1.5);
323     end;
324     if XD2(position)==0;
325         H=plot(XData,losspos2,'b');
326         hold on;
327         set(H,'LineWidth',1.5);
328     else
329         H=plot(XDatapos2,losspos2,'b');
330         hold on;
331         set(H,'LineWidth',1.5);
332     end;
333     if XD3(position)==0;
334         H=plot(XData,losspos3,'g');
335         hold off;
336         set(H,'LineWidth',1.5);
337     else
338         H=plot(XDatapos3,losspos3,'g');
339         hold off;
340         set(H,'LineWidth',1.5);
341     end;
342     hy=ylabel('Power Losses [kW]');
343     ht=title('Power Losses vs Point Load');
344 elseif ccl==1
345     if XD1(position)==0;
346         H=plot(XData,PVCCLpos1,'r');
347         hold on;
348         set(H,'LineWidth',1.5);
349     else
350         H=plot(XDatapos1,PVCCLpos1,'r');
351         hold on;
352         set(H,'LineWidth',1.5);
353     end;
354     if XD2(position)==0;
355         H=plot(XData,PVCCLpos2,'b');
356         hold on;
357         set(H,'LineWidth',1.5);
358     else
359         H=plot(XDatapos2,PVCCLpos2,'b');
360         hold on;
361         set(H,'LineWidth',1.5);
362     end;
363     if XD3(position)==0;
364         H=plot(XData,PVCCLpos3,'g');
365         hold off;
366         set(H,'LineWidth',1.5);
367     else
368         H=plot(XDatapos3,PVCCLpos3,'g');
369         hold off;
370         set(H,'LineWidth',1.5);
371     end;
372     hy=ylabel('Cumulative Cost [MR]');
373     ht=title('LCC vs Point Load');
374 elseif ati==1
375     if XD1(position)==0;
376         H=plot(XData,ATIp1,'r');
377         hold on;
378         set(H,'LineWidth',1.5);
379     else
380         H=plot(XDatapos1,ATIp1,'r');
381         hold on;
382         set(H,'LineWidth',1.5);
383     end;
384     if XD2(position)==0;
385         H=plot(XData,ATIp2,'b');
386         hold on;
387         set(H,'LineWidth',1.5);
388     else
389         H=plot(XDatapos2,ATIp2,'b');
390         hold on;
391         set(H,'LineWidth',1.5);
392     end;
393     if XD3(position)==0;
394         H=plot(XData,ATIp3,'g');
395         hold off;
396         set(H,'LineWidth',1.5);
397     else
398         H=plot(XDatapos3,ATIp3,'g');
399         hold off;

```



```

400         set(H,'LineWidth',1.5);
401         end;
402         hy=ylabel('ATT');
403         ht=title('ATI vs Point Load');
404     end;
405     hx=xlabel('Point Load [MW]');
406     set(hx,'FontSize',fsz*0.8,
407         'VerticalAlignment','top');
408     set(hy,'FontSize',fsz*0.8,
409         'VerticalAlignment','bottom');
410     set(ht,'FontSize',fsz);
411     set(gcf,'Color',[1 1 1]);
412     set(gca,'FontSize',fsz*0.7);
413     axis tight
414     a = axis;
415     AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
416     legend(namec1,namec2,namec3,0);
417     grid on;
418 end;
419
420 %Voltage Regulation/Power Losses/LCC vs Position%
421 axes(findobj(findobj('Tag','lfdcGUI','Tag','vrlyearaxes')))
422 cla;
423 if vr==1
424     H=plot(YData1,Vreg1LR,'r');
425     hold on;
426     set(gca,'Tag','vrlyearaxes');
427     set(H,'LineWidth',1)
428     H=plot(YData2,Vreg2LR,'b');
429     hold on;
430     set(gca,'Tag','vrlyearaxes');
431     set(H,'LineWidth',1)
432     H=plot(YData3,Vreg3LR,'g');
433     hold off;
434     set(H,'LineWidth',1)
435     ylabel('Voltage Regulation [%]');
436     title('Voltage Regulation vs Position');
437 elseif loss==1
438     H=plot(YData1,loss1LR,'r');
439     hold on;
440     set(gca,'Tag','vrlyearaxes');
441     set(H,'LineWidth',1)
442     H=plot(YData2,loss2LR,'b');
443     hold on;
444     set(gca,'Tag','vrlyearaxes');
445     set(H,'LineWidth',1)
446     H=plot(YData3,loss3LR,'g');
447     hold off;
448     set(H,'LineWidth',1)
449     ylabel('Power Losses [kW]');
450     title('Power Losses vs Position');
451 elseif ccl==1
452     H=plot(YData1,PVCC1LR,'r');
453     hold on;
454     set(gca,'Tag','vrlyearaxes');
455     set(H,'LineWidth',1)
456     H=plot(YData2,PVCC2LR,'b');
457     hold on;
458     set(gca,'Tag','vrlyearaxes');
459     set(H,'LineWidth',1)
460     H=plot(YData3,PVCC3LR,'g');
461     hold off;
462     set(H,'LineWidth',1)
463     ylabel('Cumulative Cost [MR]');
464     title('LCC vs Position');
465 elseif ati==1
466     H=plot(YData1,ATI1LR,'r');
467     hold on;
468     set(gca,'Tag','vrlyearaxes');
469     set(H,'LineWidth',1)
470     H=plot(YData2,ATI2LR,'b');
471     hold on;
472     set(gca,'Tag','vrlyearaxes');
473     set(H,'LineWidth',1)
474     H=plot(YData3,ATI3LR,'g');
475     hold off;
476     set(H,'LineWidth',1)
477     ylabel('ATI');
478     title('ATI vs Position');
479 end;
480 set(gca,'Tag','vrlyearaxes');
481 axis tight
482 a = axis;
483 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
484 xlabel('Distance');
485 %legend(namec1,namec2,namec3,0);
486 grid on;
487
488 if get(findobj(findobj('Tag','lfdcGUI','Tag','hrfcb'),'Value'))
489     fsz = 36;
490     fig = findobj('Tag','2dvrlposfig');
491     if isempty(fig)
492         figure
493         if vr==1
494             set(gcf,'IntegerHandle','off',...
495                 'Name','Voltage Regulation vs Position',...
496                 'NumberTitle','off',...
497                 'Tag','2dvrlposfig');
498         elseif loss==1
499             set(gcf,'IntegerHandle','off',...
500                 'Name','Power Losses vs Position',...

```

```

501         'NumberTitle','off',...
502         'Tag','2dvrlposfig');
503     elseif ccl==1
504         set(gcf,'IntegerHandle','off',...
505         'Name','LCC vs Position',...
506         'NumberTitle','off',...
507         'Tag','2dvrlposfig');
508     elseif ati==1
509         set(gcf,'IntegerHandle','off',...
510         'Name','ATI vs Position',...
511         'NumberTitle','off',...
512         'Tag','2dvrlposfig');
513     end;
514 else
515     figure(fig)
516     if vr==1
517         set(gcf,'IntegerHandle','off',...
518         'Name','Voltage Regulation vs Position',...
519         'NumberTitle','off',...
520         'Tag','2dvrlposfig');
521     elseif loss==1
522         set(gcf,'IntegerHandle','off',...
523         'Name','Power Losses vs Position',...
524         'NumberTitle','off',...
525         'Tag','2dvrlposfig');
526     elseif ccl==1
527         set(gcf,'IntegerHandle','off',...
528         'Name','LCC vs Position',...
529         'NumberTitle','off',...
530         'Tag','2dvrlposfig');
531     elseif ati==1
532         set(gcf,'IntegerHandle','off',...
533         'Name','ATI vs Position',...
534         'NumberTitle','off',...
535         'Tag','2dvrlposfig');
536     end;
537 end;
538 set(gcf,'Position',[1 29 1024 672]);
539 set(gca,'Position',[0.15 0.15 0.82 0.82]);
540 if vr==1
541     H=plot(YData1,Vreg1LR,'r');
542     hold on;
543     set(H,'LineWidth',1.5);
544     H=plot(YData2,Vreg2LR,'b');
545     hold on;
546     set(H,'LineWidth',1.5);
547     H=plot(YData3,Vreg3LR,'g');
548     hold off;
549     set(H,'LineWidth',1.5);
550     hy=ylabel('Voltage Regulation [%]');
551     ht=title('Voltage Regulation vs Position');
552 elseif loss==1
553     H=plot(YData1,loss1LR,'r');
554     hold on;
555     set(H,'LineWidth',1.5);
556     H=plot(YData2,loss2LR,'b');
557     hold on;
558     set(H,'LineWidth',1.5);
559     H=plot(YData3,loss3LR,'g');
560     hold off;
561     set(H,'LineWidth',1.5);
562     hy=ylabel('Power Losses [kW]');
563     ht=title('Power Losses vs Position');
564 elseif ccl==1
565     H=plot(YData1,PVCC1LR,'r');
566     hold on;
567     set(H,'LineWidth',1.5);
568     H=plot(YData2,PVCC2LR,'b');
569     hold on;
570     set(H,'LineWidth',1.5);
571     H=plot(YData3,PVCC3LR,'g');
572     hold off;
573     set(H,'LineWidth',1.5);
574     hy=ylabel('Cumulative Cost [MR]');
575     ht=title('LCC vs Position');
576 elseif ati==1
577     H=plot(YData1,ATI1LR,'r');
578     hold on;
579     set(H,'LineWidth',1.5);
580     H=plot(YData2,ATI2LR,'b');
581     hold on;
582     set(H,'LineWidth',1.5);
583     H=plot(YData3,ATI3LR,'g');
584     hold off;
585     set(H,'LineWidth',1.5);
586     hy=ylabel('ATI');
587     ht=title('ATI vs Position');
588 end;
589
590 hx=xlabel('Distance [km]');
591 set(hx,'FontSize',fsz*0.8,...
592 'VerticalAlignment','top');
593 set(hy,'FontSize',fsz*0.8,...
594 'VerticalAlignment','bottom');
595 set(ht,'FontSize',fsz);
596 set(gcf,'Color',[1 1 1]);
597 set(gca,'FontSize',fsz*0.7);
598 axis tight
599 a = axis;
600 AXIS([a(1)-(a(2)-a(1))*0.1 a(2)+(a(2)-a(1))*0.1 a(3)-(a(4)-a(3))*0.1 a(4)+(a(4)-a(3))*0.1])
601 legend(namec1,namec2,namec3,0);

```



```

602         grid on;
603     end;
604
605     %Contour plots of Voltage regulation/losses/Cumulative Cost
606     if ~isempty(plotc1cont) & ~isempty(plotc2cont) & ~isempty(plotc3cont)
607         axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
608         cla;
609         if vr==1
610             [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
611             hold on;
612             set(gca,'Tag','vrlcontaxes');
613             [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
614             hold on;
615             set(gca,'Tag','vrlcontaxes');
616             [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
617             hold off;
618             title('Contour plot of Voltage Regulation');
619         elseif loss==1
620             [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
621             hold on;
622             set(gca,'Tag','vrlcontaxes');
623             [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
624             hold on;
625             set(gca,'Tag','vrlcontaxes');
626             [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
627             hold off;
628             title('Contour plot of Power Losses');
629         elseif ccl==1
630             [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
631             hold on;
632             set(gca,'Tag','vrlcontaxes');
633             [cs2,h2] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
634             hold on;
635             set(gca,'Tag','vrlcontaxes');
636             [cs3,h3] = contour(XData,YData,PVCCL3,[contour*10000 contour*10000],'g');
637             hold off;
638             title('Contour plot of LCC [MR]');
639         elseif ati==1
640             [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
641             hold on;
642             set(gca,'Tag','vrlcontaxes');
643             [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
644             hold on;
645             set(gca,'Tag','vrlcontaxes');
646             [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
647             hold off;
648             title('Contour plot of ATI');
649     end;
650     set(gca,'Tag','vrlcontaxes');
651     clabel(cs1,h1)%,'labelspacing',100)
652     clabel(cs2,h2)%,'labelspacing',100)
653     clabel(cs3,h3)%,'labelspacing',100)
654     ylabel('Distance [km]');
655     xlabel('Point Load [MW]');
656     legend(namec1,namec2,namec3,0);
657
658     if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
659         fsize = 36;
660         fig = findobj('Tag','2dvrlconfig');
661         if isempty(fig)
662             figure
663             if vr==1
664                 set(gcf,'IntegerHandle','off',...
665                     'Name','Voltage Regulation Contour Plot',...
666                     'NumberTitle','off',...
667                     'Tag','2dvrlconfig');
668             elseif loss==1
669                 set(gcf,'IntegerHandle','off',...
670                     'Name','Power Losses Contour Plot',...
671                     'NumberTitle','off',...
672                     'Tag','2dvrlconfig');
673             elseif ccl==1
674                 set(gcf,'IntegerHandle','off',...
675                     'Name','Cumulative Cost Contour Plot',...
676                     'NumberTitle','off',...
677                     'Tag','2dvrlconfig');
678             elseif ati==1
679                 set(gcf,'IntegerHandle','off',...
680                     'Name','ATI Contour Plot',...
681                     'NumberTitle','off',...
682                     'Tag','2dvrlconfig');
683             end;
684         else
685             figure(fig)
686             if vr==1
687                 set(gcf,'IntegerHandle','off',...
688                     'Name','Voltage Regulation Contour Plot',...
689                     'NumberTitle','off',...
690                     'Tag','2dvrlconfig');
691             elseif loss==1
692                 set(gcf,'IntegerHandle','off',...
693                     'Name','Power Losses Contour Plot',...
694                     'NumberTitle','off',...
695                     'Tag','2dvrlconfig');
696             elseif ccl==1
697                 set(gcf,'IntegerHandle','off',...
698                     'Name','Cumulative Cost Contour Plot',...
699                     'NumberTitle','off',...
700                     'Tag','2dvrlconfig');
701             elseif ati==1
702                 set(gcf,'IntegerHandle','off',...

```

```

703         'Name','ATI Contour Plot',...
704         'NumberTitle','off',...
705         'Tag','2dvrlconfig');
706     end;
707 end;
708 set(gcf,'Position',[1 29 1024 672]);
709 set(gca,'Position',[0.15 0.15 0.82 0.82]);
710 if vr==1
711     [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
712     set(h,'LineWidth',1.5);
713     clabel(cs1,h);
714     hold on;
715     [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
716     set(h,'LineWidth',1.5);
717     clabel(cs2,h);
718     hold on;
719     [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
720     set(h,'LineWidth',1.5);
721     clabel(cs3,h);
722     hold off;
723     ht=title('Contour plot of Voltage Regulation');
724 elseif loss==1
725     [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
726     set(h,'LineWidth',1.5);
727     clabel(cs1,h);
728     hold on;
729     [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
730     set(h,'LineWidth',1.5);
731     clabel(cs2,h);
732     hold on;
733     [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
734     set(h,'LineWidth',1.5);
735     clabel(cs3,h);
736     hold off;
737     ht=title('Contour plot of Power Losses');
738 elseif ccl==1
739     [cs1,h] = contour(XData,YData,PVCCCL1,[contour*10000 contour*10000],'r');
740     set(h,'LineWidth',1.5);
741     clabel(cs1,h);
742     hold on;
743     [cs2,h] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
744     set(h,'LineWidth',1.5);
745     clabel(cs2,h);
746     hold on;
747     [cs3,h] = contour(XData,YData,PVCCCL3,[contour*10000 contour*10000],'g');
748     set(h,'LineWidth',1.5);
749     clabel(cs3,h);
750     hold off;
751     ht=title('Contour plot of Cumulative Cost');
752 elseif ati==1
753     [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
754     set(h,'LineWidth',1.5);
755     clabel(cs1,h);
756     hold on;
757     [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
758     set(h,'LineWidth',1.5);
759     clabel(cs2,h);
760     hold on;
761     [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
762     set(h,'LineWidth',1.5);
763     clabel(cs3,h);
764     hold off;
765     ht=title('Contour plot of ATI');
766 end;
767 hy=ylabel('Distance [km]');
768 hx=xlabel('Point Load [MW]');
769 set(hx,'FontSize',fsz*0.8,...
770     'VerticalAlignment','top');
771 set(hy,'FontSize',fsz*0.8,...
772     'VerticalAlignment','bottom');
773 set(ht,'FontSize',fsz);
774 set(gcf,'Color',[1 1 1]);
775 set(gca,'FontSize',fsz*0.7);
776 axis tight
777 legend(namec1,namec2,namec3,0);
778 end;
779
780 elseif ~isempty(plotc1cont) & ~isempty(plotc2cont) & isempty(plotc3cont)
781     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
782     cla;
783     if vr==1
784         [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
785         hold on;
786         set(gca,'Tag','vrlcontaxes');
787         [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
788         hold off;
789         title('Contour plot of Voltage Regulation');
790     elseif loss==1
791         [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
792         hold on;
793         set(gca,'Tag','vrlcontaxes');
794         [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
795         hold off;
796         title('Contour plot of Power Losses');
797     elseif ccl==1
798         [cs1,h1] = contour(XData,YData,PVCCCL1,[contour*10000 contour*10000],'r');
799         hold on;
800         set(gca,'Tag','vrlcontaxes');
801         [cs2,h2] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
802         hold off;
803         title('Contour plot of LCC [MR]');

```



```

804 elseif ati==1
805     [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
806     hold on;
807     set(gca,'Tag','vrlcontaxes');
808     [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
809     hold off;
810     title('Contour plot of ATI');
811 end;
812 set(gca,'Tag','vrlcontaxes');
813 clabel(cs1,h1)%,'labelspacing',100)
814 clabel(cs2,h2)%,'labelspacing',100)
815 ylabel('Distance [km]');
816 xlabel('Point Load [MW]');
817 legend(namec1,namec2,0);
818
819 if get(findobj(findobj('Tag','lfdGUI'),'Tag','hrfcb'),'Value')
820     fsize = 36;
821     fig = findobj('Tag','2dvrlconfig');
822     if isempty(fig)
823         figure
824         if vr==1
825             set(gcf,'IntegerHandle','off',...
826                 'Name','Voltage Regulation Contour Plot',...
827                 'NumberTitle','off',...
828                 'Tag','2dvrlconfig');
829         elseif loss==1
830             set(gcf,'IntegerHandle','off',...
831                 'Name','Power Losses Contour Plot',...
832                 'NumberTitle','off',...
833                 'Tag','2dvrlconfig');
834         elseif ccl==1
835             set(gcf,'IntegerHandle','off',...
836                 'Name','Cumulative Cost Contour Plot',...
837                 'NumberTitle','off',...
838                 'Tag','2dvrlconfig');
839         elseif ati==1
840             set(gcf,'IntegerHandle','off',...
841                 'Name','ATI Contour Plot',...
842                 'NumberTitle','off',...
843                 'Tag','2dvrlconfig');
844         end;
845     else
846         figure(fig)
847         if vr==1
848             set(gcf,'IntegerHandle','off',...
849                 'Name','Voltage Regulation Contour Plot',...
850                 'NumberTitle','off',...
851                 'Tag','2dvrlconfig');
852         elseif loss==1
853             set(gcf,'IntegerHandle','off',...
854                 'Name','Power Losses Contour Plot',...
855                 'NumberTitle','off',...
856                 'Tag','2dvrlconfig');
857         elseif ccl==1
858             set(gcf,'IntegerHandle','off',...
859                 'Name','Cumulative Cost Contour Plot',...
860                 'NumberTitle','off',...
861                 'Tag','2dvrlconfig');
862         elseif ati==1
863             set(gcf,'IntegerHandle','off',...
864                 'Name','ATI Contour Plot',...
865                 'NumberTitle','off',...
866                 'Tag','2dvrlconfig');
867         end;
868     end;
869     set(gcf,'Position',[1 29 1024 672]);
870     set(gca,'Position',[0.15 0.15 0.82 0.82]);
871     if vr==1
872         [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
873         set(h,'LineWidth',1.5);
874         clabel(cs1,h);
875         hold on;
876         [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
877         set(h,'LineWidth',1.5);
878         clabel(cs2,h);
879         hold off;
880         ht=title('Contour plot of Voltage Regulation');
881     elseif loss==1
882         [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
883         set(h,'LineWidth',1.5);
884         clabel(cs1,h);
885         hold on;
886         [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
887         set(h,'LineWidth',1.5);
888         clabel(cs2,h);
889         hold off;
890         ht=title('Contour plot of Power Losses');
891     elseif ccl==1
892         [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
893         set(h,'LineWidth',1.5);
894         clabel(cs1,h);
895         hold on;
896         [cs2,h] = contour(XData,YData,PVCCL2,[contour*10000 contour*10000],'b');
897         set(h,'LineWidth',1.5);
898         clabel(cs2,h);
899         hold off;
900         ht=title('Contour plot of Cumulative Cost');
901     elseif ati==1
902         [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
903         set(h,'LineWidth',1.5);
904         clabel(cs1,h);

```

```

905         hold on,
906         [cs2,h] = contour(XData,YData,ATI2,[contour contour], 'b');
907         set(h,'LineWidth',1.5);
908         clabel(cs2,h);
909         hold off;
910         ht=title('Contour plot of ATI');
911     end;
912     hy=ylabel('Distance [km]');
913     hx=xlabel('Point Load [MW]');
914     set(hx,'FontSize',fsz*0.8,...
915         'VerticalAlignment','top');
916     set(hy,'FontSize',fsz*0.8,...
917         'VerticalAlignment','bottom');
918     set(ht,'FontSize',fsz);
919     set(gcf,'Color',[1 1 1]);
920     set(gca,'FontSize',fsz*0.7);
921     axis tight
922     legend(namec1,namec2,0);
923 end;
924
925 elseif ~isempty(plotc1cont) & isempty(plotc2cont) & ~isempty(plotc3cont)
926     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
927     cla;
928     if vr==1
929         [cs1,h1] = contour(XData,YData,Vreg1,[contour contour], 'r');
930         hold on;
931         set(gca,'Tag','vrlcontaxes');
932         [cs3,h3] = contour(XData,YData,Vreg3,[contour contour], 'g');
933         hold off;
934         title('Contour plot of Voltage Regulation');
935     elseif loss==1
936         [cs1,h1] = contour(XData,YData,Ploss1,[contour contour], 'r');
937         hold on;
938         set(gca,'Tag','vrlcontaxes');
939         [cs3,h3] = contour(XData,YData,Ploss3,[contour contour], 'g');
940         hold off;
941         title('Contour plot of Power Losses');
942     elseif ccl==1
943         [cs1,h1] = contour(XData,YData,PVCL1,[contour*10000 contour*10000], 'r');
944         hold on;
945         set(gca,'Tag','vrlcontaxes');
946         [cs3,h3] = contour(XData,YData,PVCL3,[contour*10000 contour*10000], 'g');
947         hold off;
948         title('Contour plot of LCC [MR]');
949     elseif ati==1
950         [cs1,h1] = contour(XData,YData,ATI1,[contour contour], 'r');
951         hold on;
952         set(gca,'Tag','vrlcontaxes');
953         [cs3,h3] = contour(XData,YData,ATI3,[contour contour], 'g');
954         hold off;
955         title('Contour plot of ATI');
956     end;
957     set(gca,'Tag','vrlcontaxes');
958     clabel(cs1,h1)%,'labelspacing',100)
959     clabel(cs3,h3)%,'labelspacing',100)
960     ylabel('Distance [km]');
961     xlabel('Point Load [MW]');
962     legend(namec1,namec3,0);
963
964 if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
965     fsz = 36;
966     fig = findobj('Tag','2dvrlconfig');
967     if isempty(fig)
968         figure
969         if vr==1
970             set(gcf,'IntegerHandle','off',...
971                 'Name','Voltage Regulation Contour Plot',...
972                 'NumberTitle','off',...
973                 'Tag','2dvrlconfig');
974         elseif loss==1
975             set(gcf,'IntegerHandle','off',...
976                 'Name','Power Losses Contour Plot',...
977                 'NumberTitle','off',...
978                 'Tag','2dvrlconfig');
979         elseif ccl==1
980             set(gcf,'IntegerHandle','off',...
981                 'Name','Cumulative Cost Contour Plot',...
982                 'NumberTitle','off',...
983                 'Tag','2dvrlconfig');
984         elseif ati==1
985             set(gcf,'IntegerHandle','off',...
986                 'Name','ATI Contour Plot',...
987                 'NumberTitle','off',...
988                 'Tag','2dvrlconfig');
989         end;
990     else
991         figure(fig)
992         if vr==1
993             set(gcf,'IntegerHandle','off',...
994                 'Name','Voltage Regulation Contour Plot',...
995                 'NumberTitle','off',...
996                 'Tag','2dvrlconfig');
997         elseif loss==1
998             set(gcf,'IntegerHandle','off',...
999                 'Name','Power Losses Contour Plot',...
1000                 'NumberTitle','off',...
1001                 'Tag','2dvrlconfig');
1002         elseif ccl==1
1003             set(gcf,'IntegerHandle','off',...
1004                 'Name','Cumulative Cost Contour Plot',...
1005                 'NumberTitle','off',...

```



```

1006         'Tag','2dvrlconfig');
1007     elseif ati==1
1008         set(gcf,'IntegerHandle','off',...
1009             'Name','ATI Contour Plot',...
1010             'NumberTitle','off',...
1011             'Tag','2dvrlconfig');
1012     end;
1013 end;
1014 set(gcf,'Position',[1 29 1024 672]);
1015 set(gca,'Position',[0.15 0.15 0.82 0.82]);
1016 if vr==1
1017     [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
1018     set(h,'LineWidth',1.5);
1019     clabel(cs1,h);
1020     hold on;
1021     [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
1022     set(h,'LineWidth',1.5);
1023     clabel(cs3,h);
1024     hold off;
1025     ht=title('Contour plot of Voltage Regulation');
1026 elseif loss==1
1027     [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
1028     set(h,'LineWidth',1.5);
1029     clabel(cs1,h);
1030     hold on;
1031     [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
1032     set(h,'LineWidth',1.5);
1033     clabel(cs3,h);
1034     hold off;
1035     ht=title('Contour plot of Power Losses');
1036 elseif ccl==1
1037     [cs1,h] = contour(XData,YData,PVCCCL1,[contour*10000 contour*10000],'r');
1038     set(h,'LineWidth',1.5);
1039     clabel(cs1,h);
1040     hold on;
1041     [cs3,h] = contour(XData,YData,PVCCCL3,[contour*10000 contour*10000],'g');
1042     set(h,'LineWidth',1.5);
1043     clabel(cs3,h);
1044     hold off;
1045     ht=title('Contour plot of Cumulative Cost');
1046 elseif ati==1
1047     [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
1048     set(h,'LineWidth',1.5);
1049     clabel(cs1,h);
1050     hold on;
1051     [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
1052     set(h,'LineWidth',1.5);
1053     clabel(cs3,h);
1054     hold off;
1055     ht=title('Contour plot of ATI');
1056 end;
1057 hy=ylabel('Distance [km]');
1058 hx=xlabel('Point Load [MW]');
1059 set(hx,'FontSize',fsize*0.8,...
1060     'VerticalAlignment','top');
1061 set(hy,'FontSize',fsize*0.8,...
1062     'VerticalAlignment','bottom');
1063 set(ht,'FontSize',fsize);
1064 set(gcf,'Color',[1 1 1]);
1065 set(gca,'FontSize',fsize*0.7);
1066 axis tight
1067 legend(namec1,namec3,0);
1068 end;
1069
1070 elseif isempty(plotc1cont) & ~isempty(plotc2cont) & ~isempty(plotc3cont)
1071     axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
1072     cla;
1073     if vr==1
1074         [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
1075         hold on;
1076         set(gca,'Tag','vrlcontaxes');
1077         [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
1078         hold off;
1079         title('Contour plot of Voltage Regulation');
1080     elseif loss==1
1081         [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
1082         hold on;
1083         set(gca,'Tag','vrlcontaxes');
1084         [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
1085         hold off;
1086         title('Contour plot of Power Losses');
1087     elseif ccl==1
1088         [cs2,h2] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
1089         hold on;
1090         set(gca,'Tag','vrlcontaxes');
1091         [cs3,h3] = contour(XData,YData,PVCCCL3,[contour*10000 contour*10000],'g');
1092         hold off;
1093         title('Contour plot of LCC [MR]');
1094     elseif ati==1
1095         [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
1096         hold on;
1097         set(gca,'Tag','vrlcontaxes');
1098         [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
1099         hold off;
1100         title('Contour plot of ATI');
1101     end;
1102     set(gca,'Tag','vrlcontaxes');
1103     clabel(cs2,h2)%,'labelspacing',100)
1104     clabel(cs3,h3)%,'labelspacing',100)
1105     ylabel('Distance [km]');
1106     xlabel('Point Load [MW]');

```

```

1107 legend(namec2,namec3,0);
1108
1109 if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
1110     fsize = 36;
1111     fig = findobj('Tag','2dvrlcontfig');
1112     if isempty(fig)
1113         figure
1114         if vr==1
1115             set(gcf,'IntegerHandle','off',...
1116                 'Name','Voltage Regulation Contour Plot',...
1117                 'NumberTitle','off',...
1118                 'Tag','2dvrlcontfig');
1119             elseif loss==1
1120                 set(gcf,'IntegerHandle','off',...
1121                     'Name','Power Losses Contour Plot',...
1122                     'NumberTitle','off',...
1123                     'Tag','2dvrlcontfig');
1124             elseif ccl==1
1125                 set(gcf,'IntegerHandle','off',...
1126                     'Name','Cumulative Cost Contour Plot',...
1127                     'NumberTitle','off',...
1128                     'Tag','2dvrlcontfig');
1129             elseif ati==1
1130                 set(gcf,'IntegerHandle','off',...
1131                     'Name','ATI Contour Plot',...
1132                     'NumberTitle','off',...
1133                     'Tag','2dvrlcontfig');
1134             end;
1135         else
1136             figure(fig)
1137             if vr==1
1138                 set(gcf,'IntegerHandle','off',...
1139                     'Name','Voltage Regulation Contour Plot',...
1140                     'NumberTitle','off',...
1141                     'Tag','2dvrlcontfig');
1142             elseif loss==1
1143                 set(gcf,'IntegerHandle','off',...
1144                     'Name','Power Losses Contour Plot',...
1145                     'NumberTitle','off',...
1146                     'Tag','2dvrlcontfig');
1147             elseif ccl==1
1148                 set(gcf,'IntegerHandle','off',...
1149                     'Name','Cumulative Cost Contour Plot',...
1150                     'NumberTitle','off',...
1151                     'Tag','2dvrlcontfig');
1152             elseif ati==1
1153                 set(gcf,'IntegerHandle','off',...
1154                     'Name','ATI Contour Plot',...
1155                     'NumberTitle','off',...
1156                     'Tag','2dvrlcontfig');
1157             end;
1158         end;
1159     set(gcf,'Position',[1 29 1024 672]);
1160     set(gca,'Position',[0.15 0.15 0.82 0.82]);
1161     if vr==1
1162         [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
1163         set(h,'LineWidth',1.5);
1164         clabel(cs2,h);
1165         hold on;
1166         [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
1167         set(h,'LineWidth',1.5);
1168         clabel(cs3,h);
1169         hold off;
1170         ht=title('Contour plot of Voltage Regulation');
1171     elseif loss==1
1172         [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
1173         set(h,'LineWidth',1.5);
1174         clabel(cs2,h);
1175         hold on;
1176         [cs3,h] = contour(XData,YData,Ploss3,[contour contour],'g');
1177         set(h,'LineWidth',1.5);
1178         clabel(cs3,h);
1179         hold off;
1180         ht=title('Contour plot of Power Losses');
1181     elseif ccl==1
1182         [cs2,h] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
1183         set(h,'LineWidth',1.5);
1184         clabel(cs2,h);
1185         hold on;
1186         [cs3,h] = contour(XData,YData,PVCCCL3,[contour*10000 contour*10000],'g');
1187         set(h,'LineWidth',1.5);
1188         clabel(cs3,h);
1189         hold off;
1190         ht=title('Contour plot of Cumulative Cost');
1191     elseif ati==1
1192         [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
1193         set(h,'LineWidth',1.5);
1194         clabel(cs2,h);
1195         hold on;
1196         [cs3,h] = contour(XData,YData,ATI3,[contour contour],'g');
1197         set(h,'LineWidth',1.5);
1198         clabel(cs3,h);
1199         hold off;
1200         ht=title('Contour plot of ATI');
1201     end;
1202     hy=ylabel('Distance [km]');
1203     hx=xlabel('Point Load [MW]');
1204     set(hx,'FontSize',fsize*0.8,...
1205         'VerticalAlignment','top');
1206     set(hy,'FontSize',fsize*0.8,...
1207         'VerticalAlignment','bottom');

```



```

1208         set(ht,'FontSize',fsize);
1209         set(gcf,'Color',[1 1 1]);
1210         set(gca,'FontSize',fsize*0.7);
1211         axis tight
1212         legend(namec2,namec3,0);
1213     end;
1214
1215     elseif ~isempty(plotc1cont) & isempty(plotc2cont) & isempty(plotc3cont)
1216         axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
1217         cla;
1218         if vr==1
1219             [cs1,h1] = contour(XData,YData,Vreg1,[contour contour],'r');
1220             title('Contour plot of Voltage Regulation');
1221         elseif loss==1
1222             [cs1,h1] = contour(XData,YData,Ploss1,[contour contour],'r');
1223             title('Contour plot of Power Losses');
1224         elseif ccl==1
1225             [cs1,h1] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
1226             title('Contour plot of LCC [MR]');
1227         elseif ati==1
1228             [cs1,h1] = contour(XData,YData,ATI1,[contour contour],'r');
1229             title('Contour plot of ATI');
1230         end;
1231         set(gca,'Tag','vrlcontaxes');
1232         clabel(cs1,h1)%,'labelspacing',100)
1233         ylabel('Distance [km]');
1234         xlabel('Point Load [MW]');
1235         legend(namec1,0);
1236
1237     if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
1238         fsize = 36;
1239         fig = findobj('Tag','2dvrlconfig');
1240         if isempty(fig)
1241             figure
1242             if vr==1
1243                 set(gcf,'IntegerHandle','off',...
1244                     'Name','Voltage Regulation Contour Plot',...
1245                     'NumberTitle','off',...
1246                     'Tag','2dvrlconfig');
1247             elseif loss==1
1248                 set(gcf,'IntegerHandle','off',...
1249                     'Name','Power Losses Contour Plot',...
1250                     'NumberTitle','off',...
1251                     'Tag','2dvrlconfig');
1252             elseif ccl==1
1253                 set(gcf,'IntegerHandle','off',...
1254                     'Name','Cumulative Cost Contour Plot',...
1255                     'NumberTitle','off',...
1256                     'Tag','2dvrlconfig');
1257             elseif ati==1
1258                 set(gcf,'IntegerHandle','off',...
1259                     'Name','ATI Contour Plot',...
1260                     'NumberTitle','off',...
1261                     'Tag','2dvrlconfig');
1262             end;
1263         else
1264             figure(fig)
1265             if vr==1
1266                 set(gcf,'IntegerHandle','off',...
1267                     'Name','Voltage Regulation Contour Plot',...
1268                     'NumberTitle','off',...
1269                     'Tag','2dvrlconfig');
1270             elseif loss==1
1271                 set(gcf,'IntegerHandle','off',...
1272                     'Name','Power Losses Contour Plot',...
1273                     'NumberTitle','off',...
1274                     'Tag','2dvrlconfig');
1275             elseif ccl==1
1276                 set(gcf,'IntegerHandle','off',...
1277                     'Name','Cumulative Cost Contour Plot',...
1278                     'NumberTitle','off',...
1279                     'Tag','2dvrlconfig');
1280             elseif ati==1
1281                 set(gcf,'IntegerHandle','off',...
1282                     'Name','ATI Contour Plot',...
1283                     'NumberTitle','off',...
1284                     'Tag','2dvrlconfig');
1285             end;
1286         end;
1287         set(gcf,'Position',[1 29 1024 672]);
1288         set(gca,'Position',[0.15 0.15 0.82 0.82]);
1289         if vr==1
1290             [cs1,h] = contour(XData,YData,Vreg1,[contour contour],'r');
1291             set(h,'LineWidth',1.5);
1292             clabel(cs1,h);
1293             ht=title('Contour plot of Voltage Regulation');
1294         elseif loss==1
1295             [cs1,h] = contour(XData,YData,Ploss1,[contour contour],'r');
1296             set(h,'LineWidth',1.5);
1297             clabel(cs1,h);
1298             ht=title('Contour plot of Power Losses');
1299         elseif ccl==1
1300             [cs1,h] = contour(XData,YData,PVCCL1,[contour*10000 contour*10000],'r');
1301             set(h,'LineWidth',1.5);
1302             clabel(cs1,h);
1303             ht=title('Contour plot of Cumulative Cost');
1304         elseif ati==1
1305             [cs1,h] = contour(XData,YData,ATI1,[contour contour],'r');
1306             set(h,'LineWidth',1.5);
1307             clabel(cs1,h);
1308             ht=title('Contour plot of ATI');

```

```

1309         end;
1310         hy=ylabel('Distance [km]');
1311         hx=xlabel('Point Load [MW]');
1312         set(hx,'FontSize',fsz*0.8,...
1313             'VerticalAlignment','top');
1314         set(hy,'FontSize',fsz*0.8,...
1315             'VerticalAlignment','bottom');
1316         set(ht,'FontSize',fsz);
1317         set(gcf,'Color',[1 1 1]);
1318         set(gca,'FontSize',fsz*0.7);
1319         axis tight
1320         legend(namec1,0);
1321     end;
1322
1323     elseif isempty(plotc1cont) & ~isempty(plotc2cont) & isempty(plotc3cont)
1324         axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
1325         cla;
1326         if vr==1
1327             [cs2,h2] = contour(XData,YData,Vreg2,[contour contour],'b');
1328             title('Contour plot of Voltage Regulation');
1329         elseif loss==1
1330             [cs2,h2] = contour(XData,YData,Ploss2,[contour contour],'b');
1331             title('Contour plot of Power Losses');
1332         elseif ccl==1
1333             [cs2,h2] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
1334             title('Contour plot of LCC [MR]');
1335         elseif ati==1
1336             [cs2,h2] = contour(XData,YData,ATI2,[contour contour],'b');
1337             title('Contour plot of ATI');
1338         end;
1339         set(gca,'Tag','vrlcontaxes');
1340         clabel(cs2,h2)%,'labelspacing',100)
1341         ylabel('Distance [km]');
1342         xlabel('Point Load [MW]');
1343         legend(namec2,0);
1344
1345     if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
1346         fsz = 36;
1347         fig = findobj('Tag','2dvrlcontfig');
1348         if isempty(fig)
1349             figure
1350             if vr==1
1351                 set(gcf,'IntegerHandle','off',...
1352                     'Name','Voltage Regulation Contour Plot',...
1353                     'NumberTitle','off',...
1354                     'Tag','2dvrlcontfig');
1355             elseif loss==1
1356                 set(gcf,'IntegerHandle','off',...
1357                     'Name','Power Losses Contour Plot',...
1358                     'NumberTitle','off',...
1359                     'Tag','2dvrlcontfig');
1360             elseif ccl==1
1361                 set(gcf,'IntegerHandle','off',...
1362                     'Name','Cumulative Cost Contour Plot',...
1363                     'NumberTitle','off',...
1364                     'Tag','2dvrlcontfig');
1365             elseif ati==1
1366                 set(gcf,'IntegerHandle','off',...
1367                     'Name','ATI Contour Plot',...
1368                     'NumberTitle','off',...
1369                     'Tag','2dvrlcontfig');
1370             end;
1371         else
1372             figure(fig)
1373             if vr==1
1374                 set(gcf,'IntegerHandle','off',...
1375                     'Name','Voltage Regulation Contour Plot',...
1376                     'NumberTitle','off',...
1377                     'Tag','2dvrlcontfig');
1378             elseif loss==1
1379                 set(gcf,'IntegerHandle','off',...
1380                     'Name','Power Losses Contour Plot',...
1381                     'NumberTitle','off',...
1382                     'Tag','2dvrlcontfig');
1383             elseif ccl==1
1384                 set(gcf,'IntegerHandle','off',...
1385                     'Name','Cumulative Cost Contour Plot',...
1386                     'NumberTitle','off',...
1387                     'Tag','2dvrlcontfig');
1388             elseif ati==1
1389                 set(gcf,'IntegerHandle','off',...
1390                     'Name','ATI Contour Plot',...
1391                     'NumberTitle','off',...
1392                     'Tag','2dvrlcontfig');
1393             end;
1394         end;
1395         set(gcf,'Position',[1 29 1024 672]);
1396         set(gca,'Position',[0 15 0 15 0 82 0 82]);
1397         if vr==1
1398             [cs2,h] = contour(XData,YData,Vreg2,[contour contour],'b');
1399             set(h,'LineWidth',1.5);
1400             clabel(cs2,h);
1401             ht=title('Contour plot of Voltage Regulation');
1402         elseif loss==1
1403             [cs2,h] = contour(XData,YData,Ploss2,[contour contour],'b');
1404             set(h,'LineWidth',1.5);
1405             clabel(cs2,h);
1406             ht=title('Contour plot of Power Losses');
1407         elseif ccl==1
1408             [cs2,h] = contour(XData,YData,PVCCCL2,[contour*10000 contour*10000],'b');
1409             set(h,'LineWidth',1.5);

```



```

1410         clabel(cs2,h);
1411         ht=title('Contour plot of Cumulative Cost');
1412     elseif ati==1
1413         [cs2,h] = contour(XData,YData,ATI2,[contour contour],'b');
1414         set(h,'LineWidth',1.5);
1415         clabel(cs2,h);
1416         ht=title('Contour plot of ATI');
1417     end;
1418     hy=ylabel('Distance [km]');
1419     hx=xlabel('Point Load [MW]');
1420     set(hx,'FontSize',fsz*0.8,...
1421         'VerticalAlignment','top');
1422     set(hy,'FontSize',fsz*0.8,...
1423         'VerticalAlignment','bottom');
1424     set(ht,'FontSize',fsz);
1425     set(gcf,'Color',[1 1 1]);
1426     set(gca,'FontSize',fsz*0.7);
1427     axis tight
1428     legend(namec2,0);
1429 end;
1430
1431 elseif isempty(plotc1cont) & isempty(plotc2cont) & ~isempty(plotc3cont)
1432     axes(findobj(findobj('Tag','IfdcGUI'),'Tag','vrlcontaxes'))
1433     cla;
1434     if vr==1
1435         [cs3,h3] = contour(XData,YData,Vreg3,[contour contour],'g');
1436         title('Contour plot of Voltage Regulation');
1437     elseif loss==1
1438         [cs3,h3] = contour(XData,YData,Ploss3,[contour contour],'g');
1439         title('Contour plot of Power Losses');
1440     elseif ccl==1
1441         [cs3,h3] = contour(XData,YData,PVCCCL3,[contour*10000 contour*10000],'g');
1442         title('Contour plot of LCC [MR]');
1443     elseif ati==1
1444         [cs3,h3] = contour(XData,YData,ATI3,[contour contour],'g');
1445         title('Contour plot of ATI');
1446     end;
1447     set(gca,'Tag','vrlcontaxes');
1448     clabel(cs3,h3)%,'labelspacing',100)
1449     ylabel('Distance [km]');
1450     xlabel('Point Load [MW]');
1451     legend(namec3,0);
1452
1453 if get(findobj(findobj('Tag','IfdcGUI'),'Tag','hrfcb'),'Value')
1454     fsz = 36;
1455     fig = findobj('Tag','2dvrlcontfig');
1456     if isempty(fig)
1457         figure
1458         if vr==1
1459             set(gcf,'IntegerHandle','off',...
1460                 'Name','Voltage Regulation Contour Plot',...
1461                 'NumberTitle','off',...
1462                 'Tag','2dvrlcontfig');
1463             elseif loss==1
1464                 set(gcf,'IntegerHandle','off',...
1465                     'Name','Power Losses Contour Plot',...
1466                     'NumberTitle','off',...
1467                     'Tag','2dvrlcontfig');
1468             elseif ccl==1
1469                 set(gcf,'IntegerHandle','off',...
1470                     'Name','LCC Contour Plot',...
1471                     'NumberTitle','off',...
1472                     'Tag','2dvrlcontfig');
1473             elseif ati==1
1474                 set(gcf,'IntegerHandle','off',...
1475                     'Name','ATI Contour Plot',...
1476                     'NumberTitle','off',...
1477                     'Tag','2dvrlcontfig');
1478             end;
1479         else
1480             figure(fig)
1481             if vr==1
1482                 set(gcf,'IntegerHandle','off',...
1483                     'Name','Voltage Regulation Contour Plot',...
1484                     'NumberTitle','off',...
1485                     'Tag','2dvrlcontfig');
1486             elseif loss==1
1487                 set(gcf,'IntegerHandle','off',...
1488                     'Name','Power Losses Contour Plot',...
1489                     'NumberTitle','off',...
1490                     'Tag','2dvrlcontfig');
1491             elseif ccl==1
1492                 set(gcf,'IntegerHandle','off',...
1493                     'Name','LCC Contour Plot',...
1494                     'NumberTitle','off',...
1495                     'Tag','2dvrlcontfig');
1496             elseif ati==1
1497                 set(gcf,'IntegerHandle','off',...
1498                     'Name','ATI Contour Plot',...
1499                     'NumberTitle','off',...
1500                     'Tag','2dvrlcontfig');
1501             end;
1502         end;
1503     set(gcf,'Position',[1 29 1024 672]);
1504     set(gca,'Position',[0.15 0.15 0.82 0.82]);
1505     if vr==1
1506         [cs3,h] = contour(XData,YData,Vreg3,[contour contour],'g');
1507         set(h,'LineWidth',1.5);
1508         clabel(cs3,h);
1509         ht=title('Contour plot of Voltage Regulation');
1510     elseif loss==1

```

```

1511     [cs3,h] = contour(XData,YData,Ploss3,[contour contour], 'g');
1512     set(h,'LineWidth',1.5);
1513     clabel(cs3,h);
1514     ht=title('Contour plot of Power Losses');
1515 elseif ccl==1
1516     [cs3,h] = contour(XData,YData,PVCL3,[contour*10000 contour*10000], 'g');
1517     set(h,'LineWidth',1.5);
1518     clabel(cs3,h);
1519     ht=title('Contour plot of LCC');
1520 elseif ati==1
1521     [cs3,h] = contour(XData,YData,ATI3,[contour contour], 'g');
1522     set(h,'LineWidth',1.5);
1523     clabel(cs3,h);
1524     ht=title('Contour plot of ATI');
1525 end;
1526 hy=ylabel('Distance [km]');
1527 hx=xlabel('Point Load [MW]');
1528 set(hx,'FontSize',fsz*0.8,...
1529     'VerticalAlignment','top');
1530 set(hy,'FontSize',fsz*0.8,...
1531     'VerticalAlignment','bottom');
1532 set(ht,'FontSize',fsz);
1533 set(gcf,'Color',[1 1 1]);
1534 set(gca,'FontSize',fsz*0.7);
1535 axis tight
1536 legend(namec3,0);
1537 end;
1538
1539 elseif isempty(plotc1cont) & isempty(plotc2cont) & isempty(plotc3cont)
1540 axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrlcontaxes'))
1541 cla;
1542 if vr==1
1543     title('Contour plot of Voltage Regulation');
1544 elseif loss==1
1545     title('Contour plot of Power Losses');
1546 elseif ccl==1
1547     title('Contour plot of LCC');
1548 elseif ati==1
1549     title('Contour plot of ATI');
1550 end;
1551 set(gca,'Tag','vrlcontaxes');
1552
1553 if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
1554     fsz = 36;
1555     fig = findobj('Tag','2dvrlconfig');
1556     if isempty(fig)
1557         figure
1558         if vr==1
1559             set(gcf,'IntegerHandle','off',...
1560                 'Name','Voltage Regulation Contour Plot',...
1561                 'NumberTitle','off',...
1562                 'Tag','2dvrlconfig');
1563         elseif loss==1
1564             set(gcf,'IntegerHandle','off',...
1565                 'Name','Power Losses Contour Plot',...
1566                 'NumberTitle','off',...
1567                 'Tag','2dvrlconfig');
1568         elseif ccl==1
1569             set(gcf,'IntegerHandle','off',...
1570                 'Name','LCC Contour Plot',...
1571                 'NumberTitle','off',...
1572                 'Tag','2dvrlconfig');
1573         elseif ati==1
1574             set(gcf,'IntegerHandle','off',...
1575                 'Name','ATI Contour Plot',...
1576                 'NumberTitle','off',...
1577                 'Tag','2dvrlconfig');
1578         end;
1579     else
1580         figure(fig)
1581         if vr==1
1582             set(gcf,'IntegerHandle','off',...
1583                 'Name','Voltage Regulation Contour Plot',...
1584                 'NumberTitle','off',...
1585                 'Tag','2dvrlconfig');
1586         elseif loss==1
1587             set(gcf,'IntegerHandle','off',...
1588                 'Name','Power Losses Contour Plot',...
1589                 'NumberTitle','off',...
1590                 'Tag','2dvrlconfig');
1591         elseif ccl==1
1592             set(gcf,'IntegerHandle','off',...
1593                 'Name','LCC Contour Plot',...
1594                 'NumberTitle','off',...
1595                 'Tag','2dvrlconfig');
1596         elseif ati==1
1597             set(gcf,'IntegerHandle','off',...
1598                 'Name','ATI Contour Plot',...
1599                 'NumberTitle','off',...
1600                 'Tag','2dvrlconfig');
1601         end;
1602     end;
1603 set(gcf,'Position',[1 29 1024 672]);
1604 set(gca,'Position',[0.15 0.15 0.82 0.82]);
1605 if vr==1
1606     cla;
1607     ht=title('Contour plot of Voltage Regulation');
1608 elseif loss==1
1609     cla;
1610     ht=title('Contour plot of Power Losses');
1611 elseif ccl==1

```



```
1612         cla;
1613         ht=title('Contour plot of LCC');
1614     elseif atj==1
1615         cla;
1616         ht=title('Contour plot of ATT');
1617     end;
1618     hy=ylabel('Distance [km]');
1619     hx=xlabel('Point Load [MW]');
1620     set(hx,'FontSize',fsz*0.8,...
1621         'VerticalAlignment','top');
1622     set(hy,'FontSize',fsz*0.8,...
1623         'VerticalAlignment','bottom');
1624     set(ht,'FontSize',fsz);
1625     set(gcf,'Color',[1 1 1]);
1626     set(gca,'FontSize',fsz*0.7);
1627 end;
1628 end,
```

## B.4.4 Plot3d.m

```

1 function plot3d(XData,YData,namec1,namec2,namec3,UserData,Data)
2
3     vr      =   UserData(23);
4     loss    =   UserData(24);
5     ccl     =   UserData(25);
6     ati     =   UserData(26);
7
8     Vreg1   =   Data(:,1);
9     Vreg2   =   Data(:,2);
10    Vreg3   =   Data(:,3);
11    Ploss1   =   Data(:,4);
12    Ploss2   =   Data(:,5);
13    Ploss3   =   Data(:,6);
14    PVCCCL1  =   Data(:,7);
15    PVCCCL2  =   Data(:,8);
16    PVCCCL3  =   Data(:,9);
17    ATI1     =   Data(:,10);
18    ATI2     =   Data(:,11);
19    ATI3     =   Data(:,12);
20
21    XData1   =   XData(:,1);
22    XData2   =   XData(:,2);
23    XData3   =   XData(:,3);
24
25    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl1axes'))
26    cla;
27    if vr==1
28        surf(XData1,YData,Vreg1);
29        zlabel('Voltage Regulation [ %] ');
30    elseif loss==1
31        surf(XData1,YData,Ploss1);
32        zlabel('Power Losses [ kW] ');
33    elseif ccl==1
34        surf(XData1,YData,PVCCCL1);
35        zlabel('Cumulative Cost [ MR] ');
36    elseif ati==1
37        surf(XData1,YData,ATI1);
38        zlabel('ATI');
39    end;
40    set(gca,'Tag','vrl1axes');
41    view([ 60 14] )
42    shading interp
43    ylabel('Distance [ km] ');
44    xlabel('Point Load [ MW] ');
45    title (namec1);
46
47    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl2axes'))
48    cla;
49    if vr==1
50        surf(XData2,YData,Vreg2);
51        zlabel('Voltage Regulation [ %] ');
52    elseif loss==1
53        surf(XData2,YData,Ploss2);
54        zlabel('Power Losses [ kW] ');
55    elseif ccl==1
56        surf(XData2,YData,PVCCCL2);
57        zlabel('Cumulative Cost [ MR] ');
58    elseif ati==1
59        surf(XData2,YData,ATI2);
60        zlabel('ATI');
61    end;
62    set(gca,'Tag','vrl2axes');
63    view([ 60 14] )
64    shading interp
65    ylabel('Distance [ km] ');
66    xlabel('Point Load [ MW] ');
67    title (namec2);
68
69    axes(findobj(findobj('Tag','lfdcGUI'),'Tag','vrl3axes'))
70    cla;
71    if vr==1
72        surf(XData3,YData,Vreg3);
73        zlabel('Voltage Regulation [ %] ');
74    elseif loss==1
75        surf(XData3,YData,Ploss3);
76        zlabel('Power Losses [ kW] ');
77    elseif ccl==1
78        surf(XData3,YData,PVCCCL3);
79        zlabel('Cumulative Cost [ MR] ');
80    elseif ati==1
81        surf(XData3,YData,ATI3);
82        zlabel('ATI');
83    end;
84    set(gca,'Tag','vrl3axes');
85    view([ 60 14] )
86    shading interp
87    ylabel('Distance [ km] ');
88    xlabel('Point Load [ MW] ');
89    title (namec3);
90
91    if get(findobj(findobj('Tag','lfdcGUI'),'Tag','hrfcb'),'Value')
92        fsize = 36;
93        fig = findobj('Tag','3dvrlc1fig');
94        if isempty(fig)
95            figure
96            if vr==1
97                set(gcf,'IntegerHandle','off',...

```



```

98         'Name','Voltage Regulation',...
99         'NumberTitle','off',...
100        'Tag','3dvrlc1fig');
101    elseif loss==1
102        set(gcf,'IntegerHandle','off',...
103        'Name','Power Losses ',...
104        'NumberTitle','off',...
105        'Tag','3dvrlc1fig');
106    elseif ccl==1
107        set(gcf,'IntegerHandle','off',...
108        'Name','LCC',...
109        'NumberTitle','off',...
110        'Tag','3dvrlc1fig');
111    elseif ati==1
112        set(gcf,'IntegerHandle','off',...
113        'Name','ATI',...
114        'NumberTitle','off',...
115        'Tag','3dvrlc1fig');
116    end;
117    else
118        figure(fig)
119        if vr==1
120            set(gcf,'IntegerHandle','off',...
121            'Name','Voltage Regulation',...
122            'NumberTitle','off',...
123            'Tag','3dvrlc1fig');
124        elseif loss==1
125            set(gcf,'IntegerHandle','off',...
126            'Name','Power Losses ',...
127            'NumberTitle','off',...
128            'Tag','3dvrlc1fig');
129        elseif ccl==1
130            set(gcf,'IntegerHandle','off',...
131            'Name','LCC',...
132            'NumberTitle','off',...
133            'Tag','3dvrlc1fig');
134        elseif ati==1
135            set(gcf,'IntegerHandle','off',...
136            'Name','ATI',...
137            'NumberTitle','off',...
138            'Tag','3dvrlc1fig');
139        end;
140    end;
141
142    set(gcf,'Position',[1 29 1024 672]);
143    set(gca,'Position',[0.17 0.12 0.7 0.75]);
144    if vr==1
145        surfH=surf(XData1,YData,Vreg1);
146        hz=zlabel('Voltage Regulation [ %] ');
147    elseif loss==1
148        surfH=surf(XData1,YData,Ploss1);
149        hz=zlabel('Power Losses [ kW] ');
150    elseif ccl==1
151        surfH=surf(XData1,YData,PVCC1);
152        hz=zlabel('Cumulative Cost [ MR] ');
153    elseif ati==1
154        surfH=surf(XData1,YData,ATI1);
155        hz=zlabel('ATI');
156    end;
157
158    ht=title(namec1);
159    h=xlabel('Point Load [ MW] ');
160    set(h,'FontSize',fsz*0.8,...
161    'VerticalAlignment','middle');
162    h=ylabel('Distance [ km] ');
163    set(h,'FontSize',fsz*0.8,...
164    'VerticalAlignment','cap');
165    set(ht,'FontSize',fsz);
166    posz = get(hz,'Position');
167    set(hz,'FontSize',fsz*0.8,...
168    'VerticalAlignment','bottom');
169    set(gcf,'Color',[1 1 1]);
170    set(gca,'FontSize',fsz*0.7);
171
172    shading interp
173    currAxisLim = axis;
174    axis([ currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)])
175    view(60,14)
176
177    fig = findobj('Tag','3dvrlc2fig');
178    if isempty(fig)
179        figure
180        if vr==1
181            set(gcf,'IntegerHandle','off',...
182            'Name','Voltage Regulation',...
183            'NumberTitle','off',...
184            'Tag','3dvrlc2fig');
185        elseif loss==1
186            set(gcf,'IntegerHandle','off',...
187            'Name','Power Losses ',...
188            'NumberTitle','off',...
189            'Tag','3dvrlc2fig');
190        elseif ccl==1
191            set(gcf,'IntegerHandle','off',...
192            'Name','LCC',...
193            'NumberTitle','off',...
194            'Tag','3dvrlc2fig');
195        elseif ati==1
196            set(gcf,'IntegerHandle','off',...
197            'Name','ATI',...
198            'NumberTitle','off',...
199            'Tag','3dvrlc2fig');

```

```

200     end;
201 else
202     figure(fig)
203     if vr==1
204         set(gcf,'IntegerHandle','off',...
205             'Name','Voltage Regulation',...
206             'NumberTitle','off',...
207             'Tag','3dvrlc2fig');
208     elseif loss==1
209         set(gcf,'IntegerHandle','off',...
210             'Name','Power Losses ',...
211             'NumberTitle','off',...
212             'Tag','3dvrlc2fig');
213     elseif ccl==1
214         set(gcf,'IntegerHandle','off',...
215             'Name','LCC',...
216             'NumberTitle','off',...
217             'Tag','3dvrlc2fig');
218     elseif ati==1
219         set(gcf,'IntegerHandle','off',...
220             'Name','ATI',...
221             'NumberTitle','off',...
222             'Tag','3dvrlc2fig');
223     end;
224 end;
225 set(gcf,'Position',[1 29 1024 672]);
226 set(gca,'Position',[0.17 0.12 0.7 0.75]);
227 if vr==1
228     surfH=surf(XData2,YData,Vreg2);
229     hz=zlabel('Voltage Regulation [%]');
230 elseif loss==1
231     surfH=surf(XData2,YData,Ploss2);
232     hz=zlabel('Power Losses [ kW]');
233 elseif ccl==1
234     surfH=surf(XData2,YData,PVCC2);
235     hz=zlabel('Cumulative Cost [ MR]');
236 elseif ati==1
237     surfH=surf(XData2,YData,ATI2);
238     hz=zlabel('ATI');
239 end;
240
241 ht=title(namec2);
242 h=xlabel('Point Load [ MW]');
243 set(h,'FontSize',fsz*0.8,...
244     'VerticalAlignment','middle');
245 h=ylabel('Distance [ km]');
246 set(h,'FontSize',fsz*0.8,...
247     'VerticalAlignment','cap');
248 set(ht,'FontSize',fsz);
249 posz = get(hz,'Position');
250 set(hz,'FontSize',fsz*0.8,...
251     'VerticalAlignment','bottom');
252 set(gcf,'Color',[1 1 1]);
253 set(gca,'FontSize',fsz*0.7);
254
255 shading interp
256 currAxisLim = axis;
257 axis([currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)])
258 view(60,14)
259
260 fig = findobj('Tag','3dvrlc3fig');
261 if isempty(fig)
262     figure
263     if vr==1
264         set(gcf,'IntegerHandle','off',...
265             'Name','Voltage Regulation',...
266             'NumberTitle','off',...
267             'Tag','3dvrlc3fig');
268     elseif loss==1
269         set(gcf,'IntegerHandle','off',...
270             'Name','Power Losses ',...
271             'NumberTitle','off',...
272             'Tag','3dvrlc3fig');
273     elseif ccl==1
274         set(gcf,'IntegerHandle','off',...
275             'Name','LCC',...
276             'NumberTitle','off',...
277             'Tag','3dvrlc3fig');
278     elseif ati==1
279         set(gcf,'IntegerHandle','off',...
280             'Name','ATI',...
281             'NumberTitle','off',...
282             'Tag','3dvrlc3fig');
283     end;
284 else
285     figure(fig)
286     if vr==1
287         set(gcf,'IntegerHandle','off',...
288             'Name','Voltage Regulation',...
289             'NumberTitle','off',...
290             'Tag','3dvrlc3fig');
291     elseif loss==1
292         set(gcf,'IntegerHandle','off',...
293             'Name','Power Losses ',...
294             'NumberTitle','off',...
295             'Tag','3dvrlc3fig');
296     elseif ccl==1
297         set(gcf,'IntegerHandle','off',...
298             'Name','LCC',...
299             'NumberTitle','off',...
300             'Tag','3dvrlc3fig');
301     elseif ati==1

```



```

302         set(gcf,'IntegerHandle','off',...
303             'Name','ATI',...
304             'NumberTitle','off',...
305             'Tag','3dvrlc3fig');
306     end;
307 end;
308 set(gcf,'Position',[1 29 1024 672]);
309 set(gca,'Position',[0.17 0.12 0.7 0.75]);
310 if vr==1
311     surfH=surf(XData3,YData,Vreg3);
312     hz=zlabel('Voltage Regulation [%]');
313 elseif loss==1
314     surfH=surf(XData3,YData,Ploss3);
315     hz=zlabel('Power Losses [kW]');
316 elseif ccl==1
317     surfH=surf(XData3,YData,PVCCL3);
318     hz=zlabel('Cumulative Cost [MR]');
319 elseif ati==1
320     surfH=surf(XData3,YData,ATI3);
321     hz=zlabel('ATI');
322 end;
323
324 ht=title(namec3);
325 h=xlabel('Point Load [MW]');
326 set(h,'FontSize',fsz*0.8,...
327     'VerticalAlignment','middle');
328 h=ylabel('Distance [km]');
329 set(h,'FontSize',fsz*0.8,...
330     'VerticalAlignment','cap');
331 set(ht,'FontSize',fsz);
332 posz = get(hz,'Position');
333 set(hz,'FontSize',fsz*0.8,...
334     'VerticalAlignment','bottom');
335 set(gcf,'Color',[1 1 1]);
336 set(gca,'FontSize',fsz*0.7);
337
338 shading interp
339 currAxisLim = axis;
340 axis([currAxisLim(1) currAxisLim(2) currAxisLim(3) currAxisLim(4) currAxisLim(5) currAxisLim(6)])
341 view(60,14)
342 end;

```

## B.5 FUNCTION CODE TO CALCULATE THE THERMAL LOAD REACH OF THREE-PHASE LINES

### B.5.1 *Constants.m*

```

1  %Matlab routine to calculate the thermal load reach of different three-phase line technologies
2
3  %Wolf Conductor
4  S=(86.42e6);
5  pf=0.95;
6  P=(S*pf);
7  Q=(sin(acos(pf))*S);
8
9
10 r=0.2091;
11 x=0.4157;
12 b=2.72e-6;
13
14 Vs=132e3;
15 vd=0.05;
16 Vr=Vs-Vs*vd;
17
18 %Calculate constants
19 c1=Vr^4*b^2*(r^2+x^2);
20 c2=-4*Q*Vr^2*b*(r^2-b^2);
21 c3=4*(S^2*r^2+S^2*x^2-Vr^4*b*x);
22 c4=8*Vr^2*(r*P+x*Q);
23 c5=4*Vr^2*(Vr^2-Vs^2);
24
25 %Constant matrix
26 c=[c1 c2 c3 c4 c5];
27
28 %Solve matrix for all possible roots
29 roots(c)
  
```

%Deterministic or Probabilistic Thermal Load  
 %power factor  
 %real power  
 %reactive power  
 %resistance/km  
 %reactance/km  
 %susceptance/km  
 %sending-end voltage  
 %per unit voltage drop  
 %receiving-end voltage



## B.6 FUNCTION CODE TO CALCULATE THE TRANSFER CAPABILITIES OF THREE-PHASE LINES

### *B.6.1 Transfer.m*

```

1  %Wolf Conductor
2  pf=0.95;                                %power factor
3
4  r=0.2091;                               %resistance/km
5  x=0.4157;                               %reactance/km
6  b=2.72e-6;                             %susceptance/km
7  z2=(r^2 + x^2);
8
9  Vs=132e3;                               %sending-end voltage
10 vd=0.05;                                %per unit voltage drop
11 Vr=(Vs-Vs*vd);                          %receiving-end voltage
12
13 %Calculate constants
14 c1=z2;
15 c2=(2*r*pf + (sin(acos(pf)))*(2*x-b*z2))*Vr^2;
16 c3=(1-b*x+(b^2*z2)/4)*Vr^4 - Vs^2*Vr^2;
17
18 transfercapability=[c1 c2 c3];
19
20 tc=(roots(transfercapability))

```